Heikki Laaki

Virtual reality control system for industrial robot

School of Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo February 19, 2018

Thesis supervisor:

Prof. Kari Tammi

Thesis advisor:

D.Sc. (Tech.) Yoan Miche

Aalto University
School of Engineering

AALTO UNIVERSITY
SCHOOL OF ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Heikki Laaki

Title: Virtual reality control system for industrial robot

Date: February 19, 2018 Language: English Number of pages: 6+49

Department of Mechanical Engineering

Professorship: Mechatronics Code: ENG25

Supervisor: Prof. Kari Tammi

Advisor: D.Sc. (Tech.) Yoan Miche

Virtual reality (VR) is a rising technology for creating previously unseen human machine interfaces. At the same time, general purpose robotic arms are becoming more common in various use cases. The goal of this thesis is to develop a prototype system, which enables controlling a robotic arm remotely utilizing VR technology. To limit the scope, the system is intended for future research in remote medical applications such as diagnosis and surgery. Furthermore, the system will be used especially for researching safety, security and network aspects.

The developed prototype system comprises of a Universal Robots UR3 robotic arm and HTC Vive VR system. To study the functionality of the prototype system, two use cases are described and tested. In the first use case, the robotic arm is used to remotely pick up tools and move them on an operation table. In the second use case, the robotic arm is used for remote medical observations and diagnosis with a video feedback link.

The conclusion is that a VR controlled robotic system definitely has potential in medical applications. Several possibilities are already studied in this thesis, but the system would need further improvement to reach the level of safe usability required for such critical applications. Moreover, this kind of control system appears to provide unexpected possibilities, such as continuous identification and training artificial intelligence robotic systems, but further research is required. The system is a holistic combination of mechanical engineering, mechatronics and computer science, which is a source of complexity as the tools, processes and software used in different engineering areas do not always fit together well.

Keywords: Virtual Reality, Robotics, Remote Control, Teleoperation, Telesurgery

AALTO-YLIOPISTO
INSINÖÖRITIETEIDEN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Heikki Laaki

Työn nimi: Virtuaalitodellisuuteen perustuva robottikäsivarren ohjausjärjestelmä

Päivämäärä: February 19, 2018 Kieli: Englanti Sivumäärä: 6+49

Konetekniikan laitos

Professuuri: Mekatroniikka Koodi: ENG25

Valvoja: Prof. Kari Tammi

Ohjaaja: TkT Yoan Miche

Virtuaalitodellisuus (VR) on teknologia, jolla voidaan luoda ennennäkemättömiä rajapintoja koneiden ja ihmisten välille. Samanaikaisesti yleiskäyttöiset robottikäsivarret yleistyvät eri käyttötarkoituksissa.

Työn tavoite on kehittää prototyyppijärjestelmä, jossa robottikäsivartta etäohjataan virtuaalitodellisuuden avulla. Prototyyppijärjestelmän käyttötarkoitus on uudenlaisten etänä tapahtuvien lääketieteellisten sovellusten tutkimus. Erityisesti tarkoitus on tutkia turvallisuuteen, tietoturvaan ja tietoverkkoihin liittyviä ongelmia ja ratkaisuja.

Prototyyppijärjestelmä koostuu Universal Robots UR3 -robottikäsivarresta sekä HTC Vive -VR-järjestelmästä. Prototyyppijärjestelmän toimintaa tutkitaan kahden käyttötapauksen avulla. Ensimmäisessä käyttötapauksessa robotilla poimitaan ja liikutellaan työkaluja leikkauspöydällä etäohjatusti. Toisessa käyttötapauksessa robottia käytetään etädiagnoosiin ja lääketieteellisten mittausten tekemiseen videolinkin avulla.

Työn tulos on, että virtuaalitodellisuuden avulla voidaan etäohjata robottikäsivartta intuitiivisesti ja että tällaisella järjestelmällä on mahdollisuuksia lääketieteellisissä sovelluksissa. Työssä tutkitaan useita erilaisia mahdollisuuksia, mutta järjestelmä ei vielä ole riittävän kehittynyt turvallisesti käytettäväksi tällaisissa kriittisissä sovelluksissa. Lisäksi järjestelmä näyttää mahdollistavan odottamattomia lisäominaisuuksia etäohjauksen lisäksi. Esimerkiksi jatkuva käyttäjän tunnistus ja keinoälyn opettaminen ovat mahdollisia jatkotutkimusaiheita. Kehitetty järjestelmä yhdistää konetekniikkaa, mekatroniikkaa ja tietojenkäsittelytiedettä, mikä osaltaan lisää järjestelmän monimutkaisuutta, sillä eri aloilla käytetyt menetelmät, ohjelmistot ja työkalut eivät sovi aina yhteen.

Avainsanat: Virtuaalitodellisuus, Robotiikka, Etäohjaus, Etäleikkaus

Preface

My humble thanks go to Yoan Miche and others in the awesome cyber security team at Nokia Bell Labs, professor Kari Tammi and Kaisa.

Otaniemi, February 19, 2018

Heikki Laaki

Contents

1 Introduction

Virtual reality (VR) is not a new concept. Early definitions for the current understanding of VR technology date back to the beginning of 1990s. Different virtual worlds have existed in computer games and other applications for decades. In the early years of VR, it was merely a concept of fading the difference between real world and an imaginative world with some medium. [32] Some devices even tried to create the experience of immersing the user in a virtual world, but the technology was not advanced enough and the concept did not gain popularity. One example of the early VR technologies include the Forte Technology's VFX1 from 1994, which tried to make VR gaming available to the masses. [21] The VFX1 is surprisingly similar to the systems available today.

Figure 1: A person wearing an HTC Vive head mounted display (HMD) [41].

For many years, virtual realities have been explored and displayed with personal computers, game consoles and monitors. However, during the last decade, the game industry has increasingly moved to mobile smart phones and tablets [20]. Smart phones provide much freedom in creating different user interfaces by combining the various sensors included in the phones. Moreover, the data collected with smart phone sensors can be used in augmenting virtual realities with real world elements [17]. These aspects enable completely new sort of game and application development. At the same time, computer graphics technology has reached a point where virtually generated worlds are very close to reality. The best games can generate graphics and physics that can hardly be differentiated from real life videos [8]. Combining the advanced graphics and physics generation to this new type of game interfaces, the concept of VR has gained traction again [7].

Currently, VR mostly refers to a concept where the user is completely immersed into a virtual world with a head mounted display (HMD). There are other types of VR technologies as well, but in this thesis VR refers to the type using HMDs. An image of a person wearing a HMD is presented in figure 1. A HMD utilizes several sensors to follow the movement of the user's head to display content that is relative to the head's position. This creates an experience where the users really feel like they actually live inside a separate computer generated world. The technology enables possibilities outside of gaming industry as well. This thesis researches the opportunities specifically in medical applications.

The opportunities created by VR in medical applications are not obvious. However, VR technologies are already in use in advanced medical operations such as prostate cancer surgeries with a device called Da Vinci surgical robotic system shown in figure 2. The surgery is actually performed by a robot which is remotely controlled by the surgeon. The controller device is one kind of VR device which receives video and other feeds from the robot operating the patient. The surgeon then uses the device to command the robot to perform desirable maneuvers. The benefits include faster recovery because of smaller surgical wounds. [23, 34] In this kind of application, the patient and surgeon are usually situated a few meters away from each other. However, by using future networking technologies the operation can be performed over a large distance [25]. For example, with ZEUS robotic surgical system, a cholecystectomy (surgical removal of the gallbladder) was performed in 2001 over a distance of 6230 km between the surgeon and the patient [18].

Figure 2: Da Vinci surgical robotic system. Controller console on the left and the surgical robot in the middle. [28]

Nokia is working on developing 5G network technologies that will enable ultra high reliability and virtual zero latency communication. These network features are crucial in mission critical remote operations. In contrary to the dedicated connections used in remote surgeries of today, in the future, these kind of operations can be executed over mobile networks. [25]

Contrary to the specifically purpose built surgical robot and controller device such as the Da Vinci system, illustrated in figure 2, more general purpose devices are used in this thesis. A prototype system is developed using consumer grade VR devices and a general purpose industrial robotic arm. The used VR device is HTC Vive [9] combined with a gaming personal computer. The industrial robotic arm is a

Universal Robots UR3 collaborative robot [38]. The combined cost of the hardware used is below 30000 EUR which is only a fraction compared to the cost of specialized medical devices. For example, the Da Vinci system has a price tag of around 1 million USD [23, 34]. The lower cost of the hardware could make it possible to perform simulations of remote medical operations cheaper than ever. Moreover, minor medical operations, such as non-invasive examinations, could become economically worth the investment with this kind of equipment. For example, a doctor could make observations, measurements and even diagnosis on a patient remotely which could reduce costs and potentially save lives.

As the name suggests, industrial robotic arms are designed for use in factories and other industrial environments. Industrial environments have specific requirements for the hardware and specifically for the control software. However, medical applications such as the ones researched in this thesis require a different approach to controlling the arm. Therefore, special software is developed for this purpose. This special software handles the capturing of data in VR, necessary data transformations, data delivery as well as controlling the robotic arm and providing feedback to the user. Once the software required for basic functionality is developed, two use cases are described to research the possibilities and limitations of the system.

In the first use case the robot is used to move objects, such as tools, around. Moreover, a teaching sequence for assisting personnel and a feedback function is developed. In this use case the robot is also used for recording specific movements and replaying them. This functionality could potentially be used to perform some parts of the operations autonomously and the invention led to an invention disclosure within Nokia.

The second use case includes a video link back from the robotic arm. In this use case the system is used for soft operations such as observations and measurements. Different approaches are examined, such as controlling the robotic arm with the user's head or hand. Moreover, some security aspects are discussed and examined. For example, the system could be used for continuous identification of the user and even for filtering out undesired behavior and monitoring the performance level of the user. These aspects led to one invention disclosure within Nokia as well.

In section 2 the background for medical use of VR and robotic arms is discussed, following with description of the hardware in section 3 and software setup as well as the specific features of the robotic arm control in section 3.4. The developed system is tested in two use cases that are outlined in the end of section 3. The two use cases and their motivation are examined in more detail in sections 4 and 5. The first use case describes the included sub-functionalities: picking up tools in section 4.2, recording and replaying tasks in section 4.3 and force feedback in section 4.4. The second use case describes the included sub-functionalities: video feedback in section 5.2, the connectivity required for remote operation in section 5.3 and security and user identification aspects in section 5.4. The remarks made in each use case are discussed in the end of respective chapters 4 and 5.

This thesis is a part of DIMECC Cybertrust program [4] that is partially funded by Tekes. As stated on the program's website, the program creates a foundation for Finnish research and industry to address the needs emerging in the cyber security

domain. The main research objective of the DIMECC Cyber Trust program is to improve the privacy, trust and decision making in digital infrastructure by monitoring, analysing, virtualizing, and visualizing traffic, objects and events. In the program, cyber security is approached with the following themes: secure services, securing platforms and networks, advanced threats and security assurance. [4]

1.1 Goal

The goal of the thesis is to develop a prototype system which uses consumer grade VR technology to control an industrial robotic arm remotely and look at the challenges and issues raised by real-life applications of this technology. To accomplish truly remote control[1], a network connection is required between the VR and the robotic arm. The prototype system will be used in future research of safety, security and network problems and solutions in mission critical applications.

1.2 Scope

The functionality of the prototype system in medical operations is tested with two use cases. The chosen mission critical situation is medical operations. The use cases are thereby built around two different medical settings. Applications of the system outside medical operations are discussed only briefly. Moreover, the focus is to have a proof of concept and to test the performance of wireless networks in such system. Different end effectors, feedback for the user, automated use and optimization of the robot behavior are thus left out of the scope. The thesis reaches for minimal viable functionality in these areas. The safety, security and network aspects are cursorily touched.

[1] Truly remote control meaning that the controller is situated in another building or even farther from the robotic arm and that the control happens in real time with a reasonable delay.

2 Background and motivation

In this section, the background of VR, remote medical operations and industrial robotic arms is examined. Furthermore, the motivation for using these technologies in medical operations is discussed.

2.1 Virtual Reality

VR may refer to plenty of distinct concepts. Different examples of VR are described in popular culture pieces such as The Matrix film series [44]. In the movies, people are connected to a central computer and their minds are living in a simulated world, i.e. VR. Video games are often built around some kind of VR that represents or simulates real world or some real world elements. In video games, a player is able to explore and complete missions in these realities. Modern game engines such as Unity or Unreal engine are so advanced, that the virtual representation looks and behaves very realistically. At best, subjects might not be able to tell the difference between computer generated imagery and photography [8].

In this thesis, VR refers to a technology where a software generated virtual world is displayed through a HMD device. The HMD gathers data with sensors including gyroscopes, motion sensors and cameras. The gathered sensor data is processed to orientation and location information so that a computer is able to render different viewing angles based on the user head's movement. Moreover, the graphics are rendered separately for each eye of the user and as a result the user is able to perceive depth. This kind of VR technology is able to create an immersive experience where the user is actually part of the VR. The user can move and look around freely in the VR and interact with the VR world as it was real. Many users are amazed by the immersion that the system is able to generate.

Similar to the HMD, sensors can be attached to other physical objects to achieve location and rotation tracking as well. For example, a game controller can be equipped with similar technology. With a HMD and one such controller attached to each hand, a user can move and look around and manipulate virtual world objects in an intuitive manner. Moreover, passive physical objects in real world can be equipped with similar tracking device and achieve even more realistic interfaces for manipulating the VR. For example, a box with a tracking device is placed on the floor in real world. A virtual model of the box is placed on the floor in the VR. When a user moves the box in real life the virtual box model moves in the VR accordingly.

This kind of immersive VR technology is used for variety of purposes. Some examples include:

- Game industry. A player is immersed in the game world [1].
- Visualization. An architect can walk inside a building before it is built and use that as a tool for design work [5].
- Art. An artist can create 3D paintings and sculptures with an intuitive interface [6].

- Engineering. An engineer can visualize complex designs in VR during the design work [43].

- Movie industry. A movie viewer can be inside the movie or even part of the events [10].

One might realize that all the possibilities are yet to be uncovered. This thesis studies yet another purpose for VR technology.

VR technology is not completely free of problems. One common problem of immersive VR with HMDs is different sickness symptoms. The symptoms are referred to as virtual reality induced symptoms and effects (VRISE). The symptoms and effects include sickness during and after using a HMD. The causes include poor frame rate (below 60 Hz) and sensory conflicts between the visual and the vestibular senses. [30] These effects are also exposed in this thesis while testing the video feedback application in section 5.2.

VR has many benefits that are not obvious. For example, the VR environment can be manipulated manually and dynamically on the fly. Objects can float in the air and have different beneficial behaviors such as following the user. The scale of everything can be adjusted depending on the needs. Even the time can be slowed down or accelerated if needed. Some aspects are utilized in this thesis.

2.2 Remote medical operations

Medical operations often require highly specialized and trained personnel. Moreover, in many cases a person requiring medical treatment is in a critical state and delays might result in permanent traumas. Furthermore, the required personnel may be situated far away from a person in need of care. Even if there was no hurry in performing a medical operation, it might be inconvenient for the patient and the personnel to travel to a mutual location.

Thus, there is demand for remote operations in medical field. For example, it would be beneficial if patients would not have to travel long distances to meet a specialized doctor and get treatment. Such operations are possible with modern technologies. For instance, prostate cancer surgeries use a technology where a robot performs operations inside a patient while a doctor controls the operation with a special device [34]. In this case, the controller device is in the same room with the patient. Furthermore, some surgeries have already been made with patient and doctor being situated in remote locations. One example is the cholecystectomy performed over a distance of 6230 km between the surgeon and the patient in 2001 [18]. Such operations need really safe and trusted hardware, software and communication systems. Any kind of error or hacking in any part of the system might result in serious consequences for the patient.

A medical operation which utilizes remote control could be demonstrated with a consumer VR control system for an industrial robot. Such system does not have the accuracy required for actual precise medical operations. However, with a demo system various safety and security aspects can be studied and developed. The system also provides insight in the possible use cases for lighter applications such as observing and remote diagnosis. Consequently, remote medical operations and surgeries have many aspects that could benefit from the introduction of VR technology.

2.3 Industrial robotic arm

Industrial robotic arms are used for various tasks in modern manufacturing. Robotic arms are especially fit for highly repetitive work that requires accuracy for extended periods of time. At the same time, the investment required for replacing human workforce with robotic arms is substantial in terms of capital and time. Furthermore, robotic arms are programmed to a specific task and each new task requires new programming and tooling. Therefore, robotic arms are generally used in mass manufacturing as the investment can be compensated more easily. Further advantages of robotic workforce include 24/7 operation without shifts and more relaxed requirements for the working environment. [24]

Industrial robotic arms are performing tasks that are dangerous or otherwise unpleasant for human workforce. For example, lifting heavy components in an assembly line and placing them is a task that a human worker could execute only with specialized assisting equipment. Another typical use for an industrial robotic arm is painting applications. Applying an even layer of paint on a large area can be difficult and require years of experience for a human to conduct reliably and repeatedly in mass production. Moreover, the working environment can be dangerous for humans, since the chemicals used in paints can be harmful. In contrary, an industrial robotic arm can be taught to paint the layer and set to repeat the learned painting sequence [33]. Typically, the robotic arm is programmed to follow a strictly defined trajectory at a defined velocity, while spraying the paint. Consequently, the paint layer is identical between each product, quality is improved and human exposure to harmful chemicals is reduced.

Recently, robotic arms have started to work as partners with human workers in the same working space. For example, the assembly line example could be extended in a way that a robotic arm places the heavy components and a human worker fastens the components, while the robot keeps them still. Such applications require advanced safety measures from the robotic arm, since having a human in the reach of the robotic arm can result in collisions which could then cause injuries or even fatalities. Robots working with humans are specifically called cobots as in collaborative robots. Robots must comply with EN ISO 10218-1:2011 safety standard and ISO/TS 15066:2016 technical specification, which specify safety measures for cobots in industrial applications [11, 12]. To reach the required safety, systems such as collision detection can be used. However, each environment and use case is unique and the safety has to be evaluated individually before the safety level can be confirmed.

These typical described advantages of industrial robotic arms are not especially in focus of this thesis. The most important feature of industrial robotic arms utilized in this thesis is the ability to replicate human hand movement. Also other research work conducted on industrial robotic arms in medical operations have utilized this feature. For example, Mathiassen et al. have used a UR5 robotic arm in creating a ultrasound imaging system [19]. In that application, the robotic arm is used as an interface for separating the physician from the ultrasound probe. The reason for separation is that imaging with the ultrasound probe causes musculoskeletal disorders and pain for the physician, because of the high forces involved. They are using a haptic device to control the robotic arm, which is somewhat similar to the VR system used in this thesis. Remote teleoperation is also discussed in their work, but not implemented. [19]

3 Prototype system hardware and software setup

The main objective of this thesis is to use VR technology to control an industrial robotic arm and perform tasks in a medical setting. Therefore, a prototype system is developed. The specification and characteristics of all hardware components used is described in this section, followed with a description of the software setup. In the end of this section, the testing of the functionality of the prototype system is considered and consequently two use cases are outlined.

3.1 Universal Robots UR3

The prototype system is required to replicate human hand movements in three dimensions. Human hand movements are not only translations in three dimensions, but also rotations around the three axes. Thus, the physical movement replicator needs six degrees of freedom (DoF) for the replication. More rigorously, these six degrees are translations along the three axes (X,Y,Z) and rotation around each axis (yaw, pitch, roll). An industrial robotic arm with six joints is able to accomplish this [24]. Hence, the physical manipulator choice for this prototype system is an industrial robotic arm.

As discussed in the background section, industrial robots operating in the same space with humans are required to have special safety measures to avoid collisions and injuries. The prototype system is not an industrial application and therefore it does not need such measures. Nevertheless, the prototype system will be used for research and possible demonstrations with personnel unfamiliar with the system. Moreover, medical applications definitely involve humans in the same working space. For the safety of everyone involved, these safety measures would be appreciated. Building a custom robotic arm with six joints could be a possible approach to achieve the desired functionality. However, an industrial robotic arm with built in safety features for human collaboration, i.e. a cobot, provides conveniently the required functionality. Thus, the robotic arm shall be a cobot.

Repeatability	0.1 mm
Payload	3 kg
Reach	500 mm
Degrees of freedom	6 rotating joints
Collaboration operation	15 advanced adjustable safety functions

Table 1: Universal Robots UR3 specification [38]. Collaboration operation refers to the concept of human workers and a robot working in the same working space simultaneously.

Universal robots UR3 (illustrated in figure 3) is chosen after considering available models. The specification of the UR3 is presented in table 1. The DoF and safety requirements are met. Moreover, the UR3 has an extensive scripting language for building customized control software which potentially helps in achieving the desired functionality. On the minus side, the reach of the arm is slightly limited but it is sufficient for the research purposes intended for the prototype system. [39]

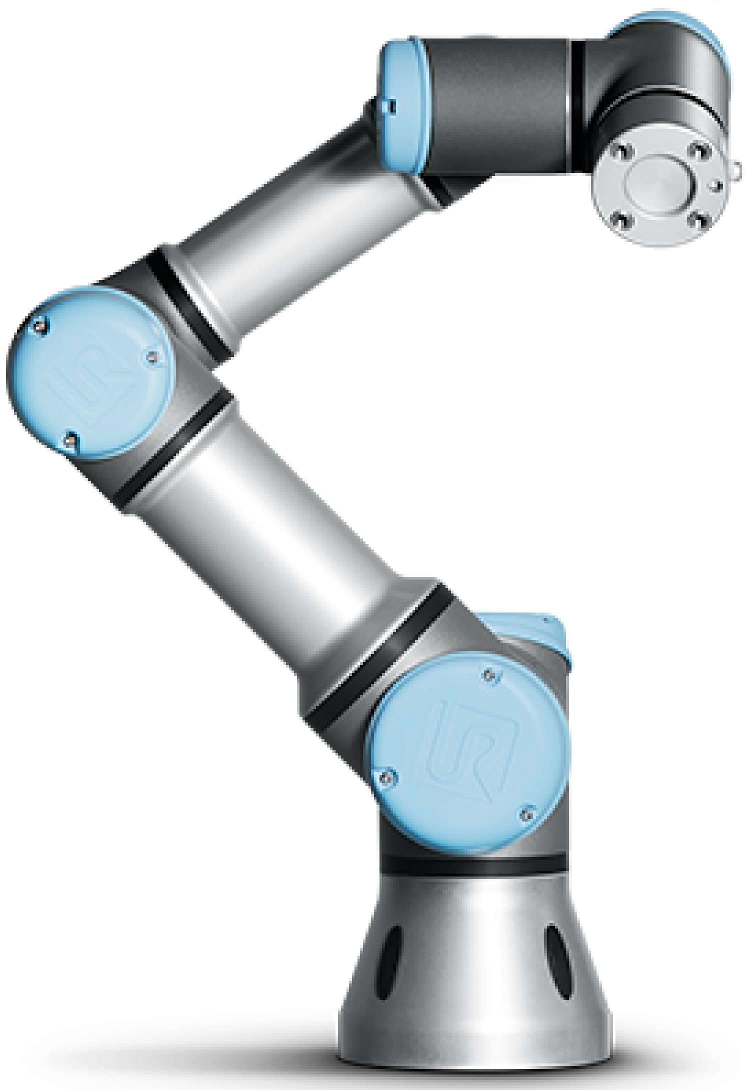


Figure 3: Universal Robots UR3 Collaborative Robot [38].

One of the safety measures embodied in the UR3 robotic arm is a force sensing functionality. The controller continuously evaluates if the arm is moving and behaving as expected or if the arm is pushing against something. If unexpected behavior or forces are detected the arm immediately performs an emergency stop to avoid the possible collision with a human. The force sensing function can also be used for other purposes. For example, in this prototype system it could serve as a source for force feedback data. Additionally, the UR3 has an interface for controlling different end effectors. An end effector provides an interface for the robotic arm to actually perform something as simply moving the arm around is not necessarily useful at all. Therefore, the interface for end effectors helps with building a system that can simulate medical operations.

3.2 HTC Vive VR system

On the VR side, the prototype system requires a possibility to display and manipulate a virtual world. Consequently, the VR system is required to contain both functionalities. The display function can be achieved with a HMD and the manipulation functionality with some sort of controller device(s).

HTC Vive (displayed in figure 4) has the required functionalities and is considered as a state of the art device of the available models based on reviews found on the internet [41]. Other possible models included Oculus Rift with its Oculus Touch controllers [26]. At the time of purchasing the equipment, the HTC Vive had better availability and ultimately was considered the best option.

Figure 4: HTC Vive HMD, two controllers and two lighthouses [9].

The Vive system includes two lighthouses that beam an infrared laser grid with spinning mirrors. The grid is utilized in achieving precise location, rotation and movement logging from the user. The HMD unit has dual AMOLED 3.6 inch diagonal screens with 1080 x 1200 pixels per eye (2160 x 1200 pixels combined), a refresh rate of 90 Hz and a 110 degree field of view. The shell of the HMD has 32 infrared detectors that are used to determine location and orientation with the laser grid. Each controller 22 similar infrared detectors, a multifunction trackpad, grip buttons, dual-stage trigger, a system button, a menu button and a haptic feedback function. [9]

The update rate of the laser location tracking system is 120 Hz, but the software provides location data at a rate of 250 Hz for the controllers and 225 Hz for the headset. The Vive system uses inertial measurement units to approximate the location at greater rate. The tracking system has a residual noise range of about 0.3 mm with two lighthouses. If only one lighthouse is used, the residual noise range increases to about 2.1 mm in the distance direction from the lighthouse. The error of the tracking over the space has a root mean square (RMS) of about 2 mm. [14] As seen when comparing the accuracy of the VR system to the accuracy of the UR3 robotic arm, the VR system is seemingly the limiting component. Although, ultimately, human precision is also a limiting factor here.

3.3 Unity and SteamVR

The VR hardware alone does not create any VR experiences and thus a software framework is required for the development of the VR scene. Unity is a platform for developing 3D games and it is widely used for creating VR experiences and games [36]. Unity handles all the tasks required to run the scenario on a personal computer such as memory management and graphics rendering. Moreover, Unity has functions for detecting collisions and providing physics that represent real life. Hence, the developer only needs to program the game logic and design the virtual world scene. Unity has a built in asset store in which every developer can distribute their work. Since there is a broad developer community around Unity, the asset store contains useful scripts for programming the game logic and virtual models for designing the scenario. [36]

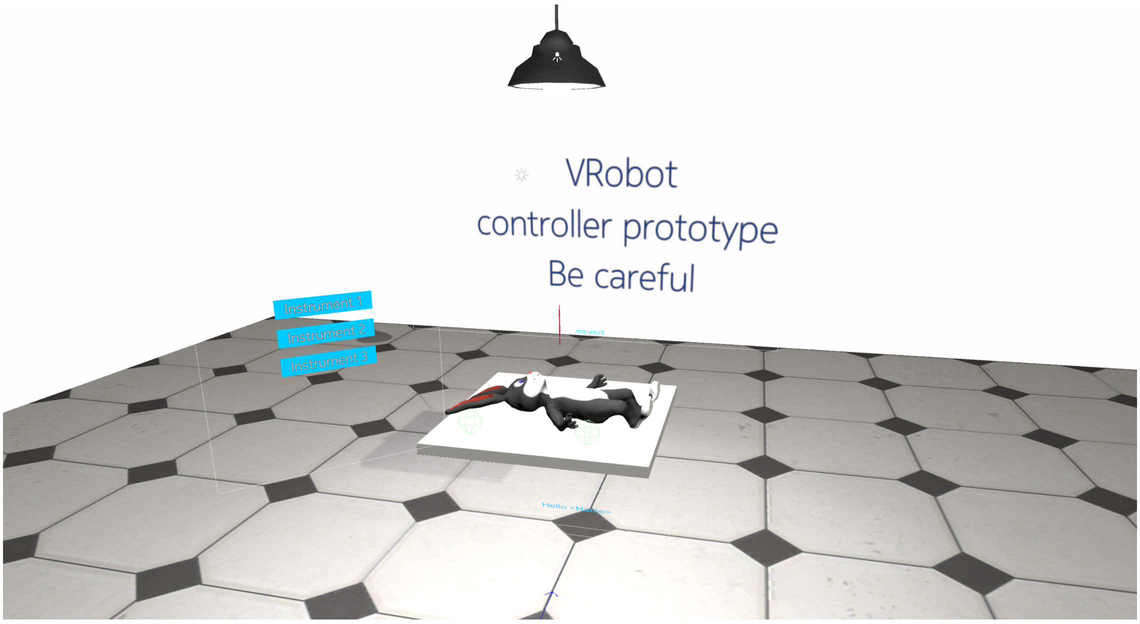


Figure 5: Virtual reality scene.

The hardware manufacturer HTC and gaming software company Valve have collaborated to create a software development kit (SDK) that provides software development tools for creating VR content and games. The SDK is called OpenVR. OpenVR contains an interface for translating the input from HTC Vive HMD and controllers to a format that Unity is able to use as well as using the haptic feedback vibrator in the controllers among other functionalities. Utilizing this SDK, the developer is not required to handle all the hardware interfaces but again can focus on creating the game logic. [40] These features resulted in the use of Unity and OpenVR for creating a scene that resembles loosely a medical setting.

There are games and medical simulator applications available that create different realistic hospital environments, but for the purpose of this thesis a simplistic environment is sufficient. Thus, a very simple scene, shown in figure 5, is built in Unity. In the scene, there is a virtual operation table in the middle. On the operation table is a virtual representation of a patient. A stuffed bunny represents a real world patient. The virtual patient is located in the middle of the operation table and the real world bunny is located so that after the conversions explained in section 3.4, the bunnies are in matching locations. Moreover, the virtual bunny is scaled up, so that when the VR movements are scaled down the movements match in relation to the patient.

Floating above the patient is a virtual representation of an instrument. This is the virtual object that is actually followed by the control loop program. To the left of the operation table are buttons for requesting a different instrument. Whenever an instrument is requested the robotic arm stops following the virtual instrument and proceeds to return the current instrument to its storage position and picks up the requested instrument. This is explained more in section 4.2.

The scene is a simplified version of a possible medical operation scenario. Since the scenario is completely virtual all features can be changed manually or with a script depending on the situation and the doctor's needs. The possibilities of VR are discussed more in section 2.1.

3.4 Control loop

Once the robotic arm and VR equipment are set up, a communication link between these two systems is needed. This section examines the calculations and restrictions required as well as the actual program that handles the data delivery between the VR scene and the robotic arm.

The UR3 controller has an Ethernet port which is used to link the robotic arm controller to a PC running the VR scene. This setup is used only for the development phase and is developed further after the control loop design is working. The ethernet cable connectivity is replaced with a 4G connection in order to have a truly remote operation in a later phase of the work. The remote connectivity and its details are examined in section 5.3.

When using the system, a user moves a virtual instrument in the VR world with the help of a controller in their hand. The VR hardware tracks the movement of the controller and delivers the location and rotation data to the VR scene. The location and rotation is interpreted in the OpenVR software layer and the result is location and rotation data relative to the VR scene. Using that data, the system is able to tell the robotic arm what pose it should reach for. However, the coordinate system used in the VR differs from the coordinate system used in the robotic arm control system. Moreover, the system has other limitations that must be accounted for, such as the reach of the robotic arm. Consequently, several translations and other conversions have to be applied before the robotic arm can use the data.

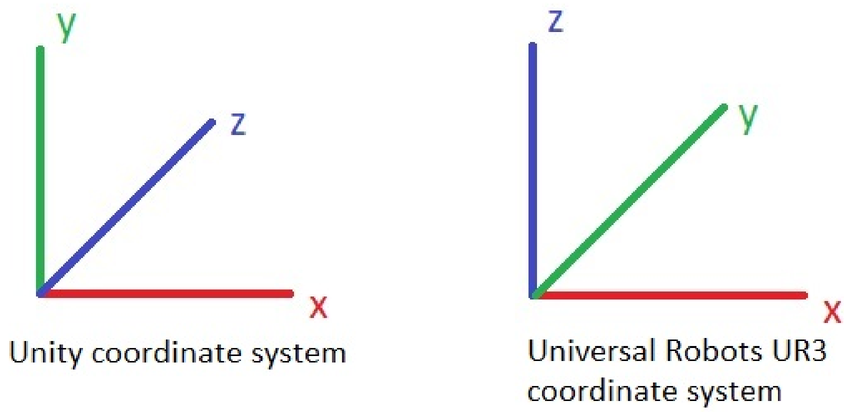


Figure 6: The coordinate systems of Unity engine and Universal Robots controller.

The Unity based VR world has a left handed coordinate system with X-axis to the right, Y-axis up and Z-axis to forward. Furthermore, the location and rotation in the VR software are stored in a quaternion format. On the contrary, the robotic arm controller uses a coordinate system with X-axis to the right, Y-axis to forward and Z-axis upward. The rotation in the robotic arm controller is stored in an axis-angle vector [37]. The coordinate systems are illustrated in figure 6. To achieve a replication of the movement a change of basis is required first:

$$\begin{cases} X_r = X_{vr} \\ Y_r = Z_{vr} \\ Z_r = Y_{vr} \end{cases} \tag{1}$$

Where the X_{vr}, X_{vr} and X_{vr} are the coordinates in the VR scene and X_r, Y_r and Z_r the coordinates that are delivered to the robot controller.

Next, the rotation data has to be translated from quaternion format to an axis-angle vector format:

$$\begin{cases} \alpha = 2 * acos(qw) \\ x_{vr} = qx/\sqrt{1 - qw * qw} \\ y_{vr} = qy/\sqrt{1 - qw * qw} \\ z_{vr} = qz/\sqrt{1 - qw * qw} \end{cases} \tag{2}$$

Where α is the angle in the axis-angle format. x_{vr}, y_{vr} and z_{vr} are the rotation axis components (in the VR scene coordinate system). qx, qy, qz and qw are the components of the rotation quaternion.

The UR3 controller stores the angle component as the length of the axis vector [37]. Thus, the resulting axis has to be normalized and multiplied by the angle from equation 2, which is done in equations 3 and 4, where l is the length of the original vector:

$$l = \sqrt{x_{vr}^2 + y_{vr}^2 + z_{vr}^2} \tag{3}$$

$$\begin{cases} x_{vr} \leftarrow x_{vr} * \alpha/l \\ y_{vr} \leftarrow y_{vr} * \alpha/l \\ z_{vr} \leftarrow z_{vr} * \alpha/l \end{cases} \tag{4}$$

Finally the basis has to be changed, because of the different coordinate system:

$$\begin{cases} x_r = -x_{vr} \\ y_r = -z_{vr} \\ z_r = -y_{vr} \end{cases} \tag{5}$$

Where x_r, y_r and z_r are the components of the axis-angle vector in the robot arm coordinate system with the angle as the length of the vector.

These conversions would be sufficient for a successful exact replication of the movements in the VR. However, the instrument location in the VR scene is measured in relation to the virtual operation table. More explicitly, when the instrument is in the middle of the operation table the measured location would be in the middle of the robot coordinate system. The robotic arm itself is located in the center of the coordinate system and thus cannot operate close to that area, because it would collide with itself. Since the robotic arm is mounted directly to the operation table, the center of the coordinate system has to be offset to avoid the robotic arm colliding. Therefore, an offset value has to be added to the robot Y-coordinate Y_r.

For actual usability, research and demonstrative purposes the robotic arm cannot stand in front of the operation table. Otherwise observing or assisting the robotic arm operation would be blocked by the arm itself. A more favorable position for the robotic arm would be behind the operation table. Thereby free operation space for assisting work and observing the robot working would be achieved. Consequently, the offset would be best in negative direction. The offset is applied in equation 6. Adding this to the equation 1, the resulting coordinates to be delivered to the robot are shown in equation 7. The robot operates hereby as if the doctor was standing in front of the operating table while the robot is standing behind the operation table.

$$Y_r = Z_{vr} - s_{offset} \tag{6}$$

$$\begin{cases} X_r = X_{vr} \\ Y_r = Z_{vr} - s_{offset} \\ Z_r = Y_{vr} \end{cases} \tag{7}$$

3.4.1 Safety box

As discussed in the hardware section 3.1, the robotic arm has a slightly limited reach of 0.5 m measured from the middle of the robot base among other restrictions. More specifically, in this application, the constraints can be formulated as follows:

- The robot cannot operate outside of the spherical reach area at all.
- The robot operation is limited close to the edge of the spherical reach area.
- The robot cannot operate close to the robot base.

- The robot cannot operate below the operation table.

To address these constraints, a so called safety box is created. The safety box is implemented so that if the user moves the virtual instrument out of the box in the VR world, the system ignores coordinates that are out of the limits and replaces them with values within the limits. Fitting this safety box inside the constraints listed above results in a very limited operating space. Suitable dimensions for this safety box in this implementation are width 0.4 m, depth 0.3 m and height 0.25 m. However, in the VR world there are significantly fewer constraints. Actually one of the major benefits of the system lies within the freedom to design the VR scene. The movements in the VR world can be manipulated in many ways before delivering them forward. In this implementation, the movements in VR are simply scaled down with a factor of 0.5.

By combining the calculations explained and the safety box a conversion between VR and robotic arm coordinate systems is achieved. It takes into account:

- differences in coordinate systems.
- differences in rotation representation systems.
- convenient location of the robotic arm on the operation table.
- restrictions of the robotic arm.

These conversions are executed before delivering the data to a separate server software as an ASCII encoded message. The actual programming of the system is described in the next section.

3.4.2 Control loop program

In this section, the program code of the system and its functionality is examined. Moreover, the data flow between all components of the system is detailed. The system is separated in two sites, the VR site and robot/patient site. Both sites are running a server computer that handles the networking and most of the data processing. The VR site computer runs also the VR scene. A robot controller handles the processes that are closely related to the robotic arm. A schematic of the data flow is shown in figure 7. A simplified network schematic can be found in figure 18.

The parts of the software that are related only to the VR scene functionality are left out of this thesis. Only the parts that contribute to the functionality of the control are represented. The description begins from the VR site scene program. The program pseudocode is displayed in program 1. The program first gets the location and rotation data from the VR scene. Next, the safety box function and the conversions explained in previous section are applied. Then the data is sent to the VR server program running on the same computer. In the end of the loop, the program receives force feedback data and drives the haptic vibration device

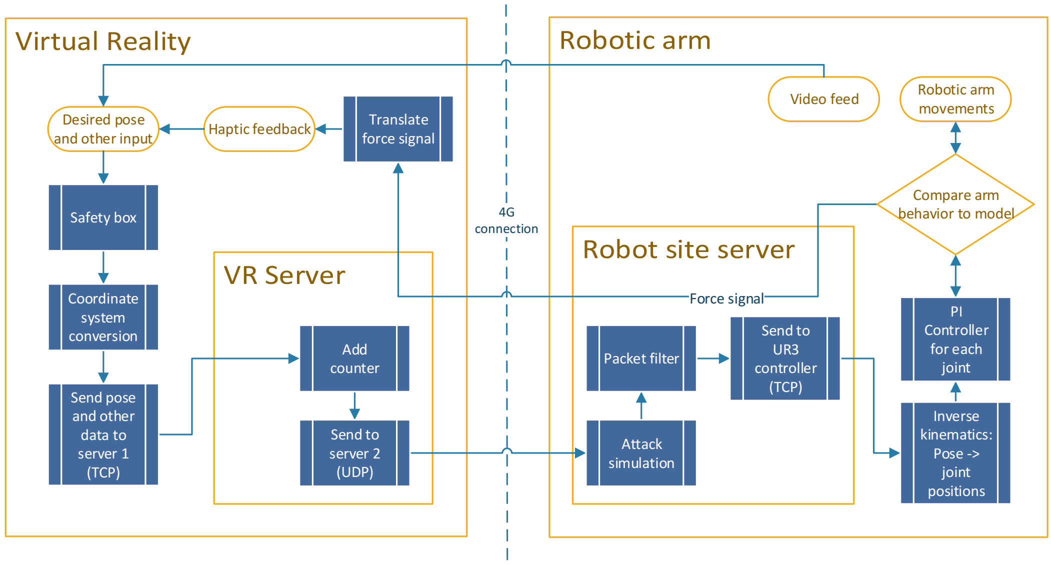


Figure 7: Data flow diagram.

in the hand held controller with the force feedback data. As a result, the hand-held controller vibrates in proportion to the applied force. The communication is conducted with TCP inside each site.

while *forever* **do**
    get instrument location;
    get instrument orientation;
    apply safety box;
    unity $\rightarrow$ robot coordinate conversion;
    quaternion $\rightarrow$ axis-angle conversion;
    send to server (location + rotation) with TCP;
    receive force data with TCP;
    set haptic feedback(force);
end

Program 1: VR scene code

The VR server program displayed as program 2 parses the message from the VR scene and adds a counter to the beginning. The counter is a running number that is used for packet delivery checking. The detailed functionality of the network is explained in section 5.3. The message is forwarded to the robot/patient site server program over a 4G connection using UDP. Then the force feedback data from the robot/patient site is received and then delivered back to the VR scene using TCP.

```
counter ← 0;
while forever do
    receive message from unity with TCP;
    send message (counter + message) to filter with UDP;
    counter ← counter + 1;
    receive force data;
    send force data to unity with TCP;
end
```

Program 2: VR server code

The robot/patient site server program (program 3) receives the message and parses the counter number. In case the running counter number of a message is equal or less than the number of a previously received message, the system filters out that message completely. Otherwise the message is forwarded to the robot controller via an ethernet cable using TCP. Again, the force feedback data is delivered to the VR server program using UDP.

```
counter ← 0;
while forever do
    message ← message from VR server;
    if message counter > counter then
        send message to robot with TCP;
        counter ← message counter;
    end
    receive force data with TCP;
    send force data to VR server with UDP;
end
```

Program 3: Filter server code

The program that controls the robotic arm and its interfaces is shown in program 4. The controller has a control loop that runs every 8 ms or at a rate of 125 Hz and the robotic arm has to be given instructions once each run. Consequently, the robotic arm control program is split in two threads. Thread 1 provides the instructions to the robotic arm, to avoid missing the timeframe for giving the instructions. Thread 2 in turn handles the data flow and updates the desired rotation and location whenever new data is received. Thread 2 also reads the estimated force data and delivers it to the server program using TCP. The force estimation is explained in section 4.4. The two threads are synced to ensure robust functionality. Consequently, every run of the loop reads the message from the robot/patient site server program. The message is parsed and the robotic arm controlled accordingly if the received data indicates that the robotic arm needs to do something.

The UR3 programming language has built-in functions for moving the arm to a desired pose. A pose is a combination of exact position and orientation. However, the built-in move functions do not allow a dynamically updating goal pose, such as the one that the thread 2 is providing. Using these built-in move functions results in trembling movement as the arm moves to a desired pose and stops before proceeding to the next desired pose. Moreover, the arm continuously falls behind from the real time input and the controller queues the requests.

To improve the behavior, the built-in functions have to be replaced. Ravn et al. have noticed that a dynamically updating desired pose can be achieved only by controlling the joint servo motors directly [27]. This is accomplished with another built-in function called inverse kinematics.

```
1  /* THREAD 1 */
2  while forever do
3      get current joint position;
4      inverse kinematics(desired pose);
5      calculate difference between desired and current joint positions;
6      multiply difference with gain;
7      if tool changed then
8          return current tool;
9          pick new tool;
10     end
11     if magnet state changed then
12         magnet switch;
13     end
14     if safety ok then
15         joint speed(difference, acceleration, time step);
16     else
17         move to safe home position;
18     end
19 end
20 /* THREAD 2 */
21 while forever do
22     receive message from server with TCP;
23     parse desired pose, magnet state and safety ok data;
24     send estimated force data to filter with TCP;
25     sync the threads;
26 end
```

Program 4: Robot control code

The location and rotation data read from the received message is transformed to an angular desired position for each of the robot joints (program 4 line 4). With the desired joint positions, a PI control loop is created. The control loop reads the current angular position of each joint (line 3). The current position of each joint is subtracted from the desired angular position (line 5). The difference is multiplied with a gain factor (line 6). The multiplied value is used as a input signal for a speed controlling function. Thus, each joint receives a command to move to a desired direction with a desired speed. The speed control function also uses defined acceleration and time step, which can be used for tuning the behavior of the control loop along with the gain factor. The robot control program handles other control signals as well, such as initiating the tool change process, the end effector interface or blocking the robot operation in case of compromised security. All these input signals are updated once at each run of the control loop or at 125 Hz.

The design of the control loop makes the arm move smoothly. However, the loop also causes the arm to lag behind the desired pose in case of really fast movement. Moreover, as some joints are able to move faster than others, the trajectory is not exactly as desired with fast movements. For example, if the user tries to move the arm rapidly along a straight line the arm moves along a slightly curved path that is close to the desired straight line. Partially this is because the gain factor used is equivalent for every joint. A unique optimized gain factor for each joint would improve the arm behavior further, because the typical movements are very different for the bigger joints close to the robot base compared to the joints towards the end of the robotic arm.

A video demonstration of the control loop functionality is available at [16] and a screen capture of the video in figure 8. The described features of the control loop can be distinguished in the video.

Figure 8: A video demonstration of the robot tool-head tracking the movement of the hand-held controller [16] https://youtu.be/Sfv-Mot8cvo.

Nonetheless, in the scope of this thesis, the ultimate optimization of the behavior of the system would have been unreasonable. Tuning a PI-controller for a single joint alone is a task that requires special tools and knowledge. A six DoF robotic arm has six joints and in this kind of use case the behavior of each joint varies greatly from each other. Furthermore, replicating human hand movements involves a great variety of different movements which adds even more complexity in tuning the controller. Therefore, the ultimate tuning of the controller factors would be a subject for another research work. The gain factor used in this system is a result of trial and error. The behavior is acceptable in most situations and thus a viable operation level is reached.

There are some unfavourable locations in the robot operating space especially close to the robot base. When the robotic arm is operating close to its base, the robotic arm is "folded" i.e. the first stretch of the arm protrudes away from the base and the next reaches back towards the base. Within these areas even small movements of the end of the robot would require some joints to move significant distances. If the speed of the movement is high at the same time, the joints are required to move at speeds that exceed their specification. Partially this is taken care of in the system, because the safety box function forces the operation area away from the robot base. However, since the reach of the arm is limited, part of the safety box area is still relatively close to the robot base, and movements in that part cause more error in the trajectory, especially with higher velocities.

3.5 Outline for 2 use cases

The basic functionality i.e. the replication of hand movements was established in the first phase of the work. Going forward, the control system has to be tested and researched. One possibility would have been to go into the fine details of the system and try to measure its limits and capabilities. However, since the system is at prototype level and the technology around VR is advancing in an accelerated pace, that might result in minor outcomes. On the contrary, the system as a concept is a limited representation of the possibilities around this kind of machine control. Hence two use cases are conceptualized to demonstrate and research nontrivial advantages this system could introduce especially in medical field. The two use cases are outlined here and examined in detail in later sections.

3.5.1 Tool pickup and teaching a robot

It is obvious by now that the robot is used to mimic a human performing some kind of medical operation. However, following the movement of the operator in VR does not really contribute to performing anything. Hence a interface for picking up tools and using them is created and the use case is built around that.

In the use case the robot is used to pick up a tool and move it since many medical operations require tools and other equipment. This kind of robot application would be used to actually perform an operation on a patient. Possible examples of these medical operations include injections, surgeries and different measurements.

Moving a tool around and performing a task with it seems simple. However, a robot does not have any way to pick up anything without an interface. Within robotics this interface is called end effector [24]. Furthermore, once the pickup interface is implemented the robot has to be taught how to use the picked up equipment. The UR3 has a versatile interface for different end effectors. However, in the scope of this research the end effector is not in the focus. A simple electric magnet is

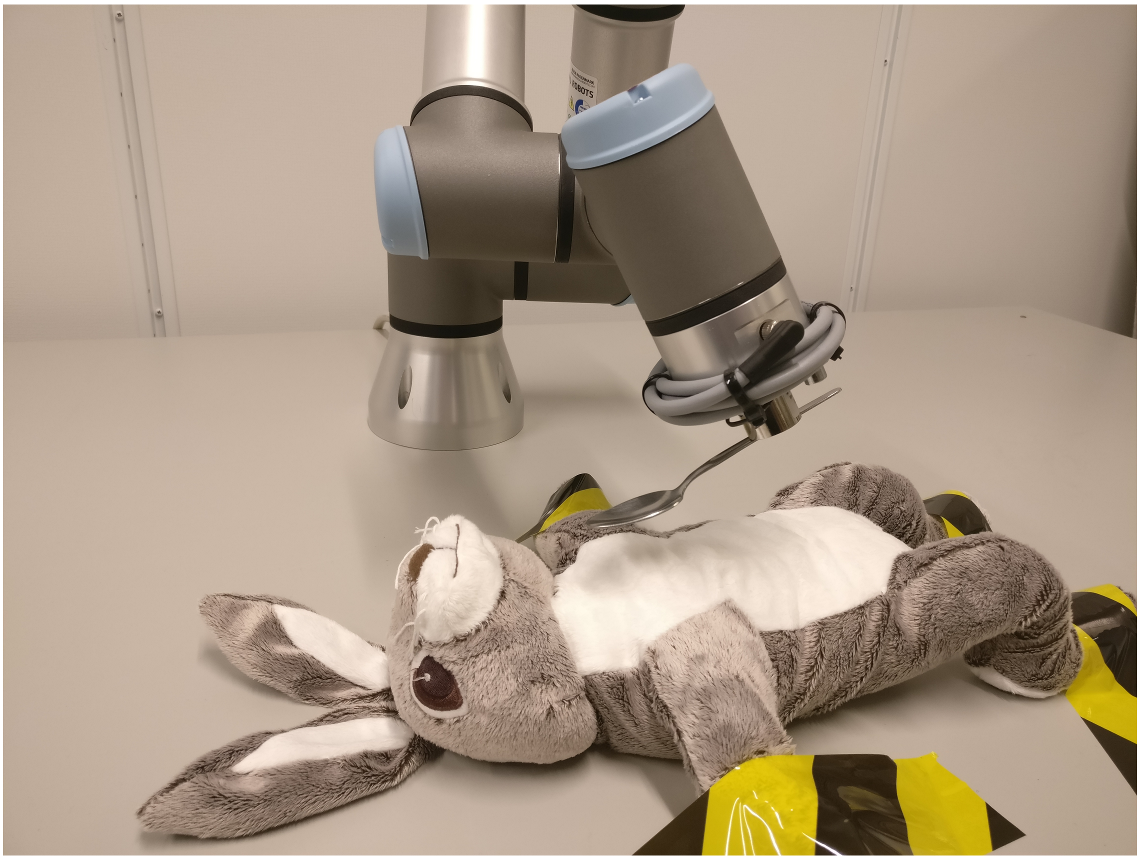


Figure 9: The robotic arm holding a tool above a patient.

attached to the end of the UR3 arm to achieve pick up functionality. An example of the robot holding a tool is shown in figure 9. The user is able to switch the magnet on and off with the controller in VR. The tools are attached to a piece of metal that sticks to the magnet once it is switched on. Furthermore, the robot control program could also operate the magnet for picking up equipment on its own. This is used for example to command the robot to pick up another tool in the middle of a operation. Also a teaching sequence is developed for teaching the controller pickup locations of tools and other equipment.

As discussed, the UR3 robot has a function for estimating the force that the robot applies to the environment. A human performing operations is relying on "force feedback" naturally when using tools in real life. However, when controlling a robot in VR any natural force feedback is not felt and the control feels very numb. Hence, the force estimation is used to develop a simple force feedback function. The user feels a vibration in the controller when the UR3 arm touches something. The intensity of the vibration is dependent to the force applied. A user eventually learns how the vibration represents the force applied and the feedback helps in achieving accurate operation. The details of the use case, tool pickup function and force feedback are described and discussed in section 4.

As discussed in the hardware section 3.1, best use for a robot would be automation of every task. To research this, the control system is used to perform a task in this use case and the control movements and commands are recorded. The recorded control sequence is then used for repeating the task without human control. Furthermore, the recorded sequences could be used for teaching an artificial intelligence to perform similar tasks autonomously in slightly varied situations. This research resulted in an invention disclosure within Nokia.

3.5.2 Video application

Many medical operations do not actually require physically performing complex operations on the patient. The doctor might, for example, need to examine the patient in order to make a diagnosis before writing a prescription for medication.

The second use case is built to represent this kind of lighter medical operations such as observations and measurements. To achieve this functionality, the user needs to get data back from the patient site. In this case, the robotic arm is equipped with a video camera to get visual feedback data. The video camera provides a video feed that is sent to the remote VR world.

With the VR control system a user is now able to look around in the physical world. In the first use case where the user would operate with some kind of tool, it is natural that the user controls the robot with their hands. Observing a patient remotely might need different means of control. Therefore, two different control versions were tested. The first version moves the arm with the hand controller similar to the first use case. Controlling the position and angle of the camera might work very well for some applications where precision is needed. However, moving the viewing angle with your hand does not necessarily work for operations where situational awareness is required. Hence, the second version moves the robotic arm and the camera in relation to the user's head. This results in a more natural way of looking around on the operation table.

Moving the video camera around with hand proves to be an intuitive way to make observations. Similar to the tool movement case, the user moves a virtual instrument around to tell the robotic arm where to point the camera. The video feed is displayed on a floating screen in the VR scene. The user has an option to fix the screen to be static or to move following the hand of the user. In the second version, the robotic arm follows the head of the user. The video feed is displayed right in front of the user's eyes which creates a feeling that the user actually looks around in the real world. The video feed was in offset angle compared to the users head in the VR world. This misalignment caused nausea and even slight vertigo in some users.

All the communication between VR and the robotic arm controller happens over a 4G connection to evaluate the system functionality in real life remote operation. With optimal operating conditions the communication is really smooth. A relatively stable delay of roughly 0.2 seconds is present while using the system. The communication is also exposed to various simulated network attacks to reveal the effects on the control system. For example, a simple denial of service attack causes the robotic arm to shake to an extent that makes the system unusable.

Using VR system to record natural movements of a user provides some unique data points that can't be accessed normally. For example, the location of the user's hands and head can be measured in relation to each other or to other objects. Specific analysis on that data can reveal, for example, if the user's hands are shaking abnormally which could suggest that the user is, for instance, tired or under influence of alcohol. This invention is used to research possible novel user identification and security features and resulted in an invention disclosure as well.

4 First use case: remote medical operation with a tool

In this section, the first use case is described in detail. A detailed description of each subfunctionality is provided as well as analysis on the system performance. Furthermore, based on the experiences in this use case, potential additional applications outside medical field are proposed.

4.1 Motivation

As already discussed, performing an operation frequently requires the ability to pick up and move tools and other objects. In the first use case, the robot is equipped with a simple electric magnet for picking up tools. When the robot is able to pick up and use a selection of tools, the system can be used for performing tasks with wide variety.

In medical context, examples of such tasks vary from simple measurements to cutting the skin and performing invasive operations. For example, the robot could pick up a stethoscope which sends the audio to the human operator in the VR. The current state of the art devices use this kind of functionality for minimally invasive surgeries. A doctor controls a robot with a user interface and the robot performs tasks inside the patient. The possible tools used include for example needles, blades and brushes. The operation relies mostly on visual feedback from a camera usually called endoscope. With the feedback the doctor knows what kind of input is needed to accomplish the task at hand. [34, 23]

In this use case, the tools used are placed on the operation table. An example set of tools can be seen in figure 11d. A special teaching sequence is used to teach the robot where each tool is located. In this case, an assistant moves the robotic arm by hand to each pickup location and then commands the control program to save that location. Later, during the actual operation, the control program uses these saved locations to pickup the tools the user needs.

The VR control system could be easily used for performing a task while recording all the input the operator provides. With the recorded sequence, a similar task could be performed without a human operator. A recorded operation might also suit for training inexperienced doctors and even artificial intelligence. This kind of functionality introduces some unique possibilities and requirements discussed in detail in section 4.3.

The use case is focusing on medical field but the functionality might potentially have applications in various other industries. Traditionally robots are used for highly repetitive work. The benefits come from low error rate, faster working speed and capability to work without getting tired [24]. Contrary to that, in this use case, the robot is used as an accurate replacement for a human operator but the actual operator is a human. The tasks that the system is designed to perform are roughly repetitive from one time to another, but with live patients they are not exactly similar each time. Thus, the traditional benefits of robot workforce are not actually exploited. The benefits of the system in medical field are discussed in section 4.5.

The following sections examine the subfunctionalities beginning with picking up tools.

4.2 Tool pickup and teaching sequence

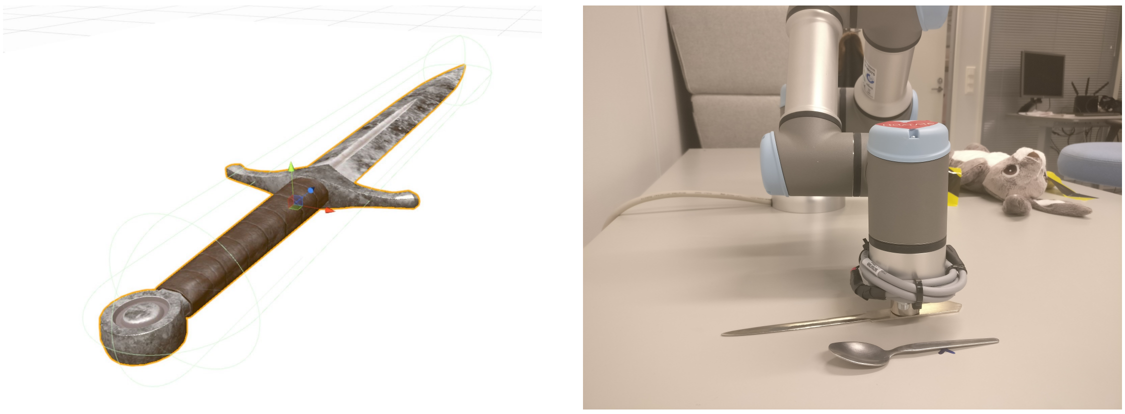
Picking up a tool and using it to accomplish a task is a simple functionality for a human. However, an industrial robot is not designed to use several tools by default. More rigorously, a robot could be equipped with a single tool to repeat a certain set of actions endlessly. To be able to pick up different tools without human intervention, the robot requires an interface to attach and detach tools and use them. In industrial applications, the robot usually has a standardized interface to achieve this and the different tools are designed to attach to the interface. All the tools must be measured and the dimensions and weight saved in a database. This way the robot knows how to pick up a tool, as well as the orientation and effective dimensions and weight of the tool.

In this application the robot is required to pick up different tools and possibly other objects depending on the situation. The solution in industrial applications would be a gripper end effector such as the Weiss Robotics gripper [45] shown in figure 10. A gripper is a versatile attaching interface which does not necessarily require exact dimensions of the object that is picked up. However, using standardized industrial attaching interfaces would be excessive, because those are designed for usage in mass production factories. A gripper end effector would have been optimal for this research but it would have added more complexity and price to the hardware. Thus, the end effectors were decided to be left out of the scope of this research and a simpler solution was chosen.

Figure 10: Weiss Robotics gripper end effector [45].

As discussed above, a versatile but cost effective tool attachment solution is required. An electric magnet is a simple interface to hold, pick up and release metallic objects and seemed to suit well in this application. The chosen electric magnet is capable of holding a load of 5 kg which exceeds the lifting capability of the UR3 robotic arm (3 kg). All the tools the robot uses must be attached to a piece of metal that attaches to the electric magnet. Now, the robot is able to pick up tools and use them for operations. However, there is still the problem of tool orientation and dimensions as with the standardized industrial interfaces.

(a) Carefully placing the tools. (b) Moving the arm to a pick up location.



(c) A matching 3D model of a tool. (d) Robot picking up a tool.

Figure 11: Phases of the teaching sequence.

To overcome this problem, a teaching sequence is developed. The teaching sequence has to be completed before the robot is able to perform anything. First, a human assistant, situated at the robot site, places the tool objects on the operating table carefully in a precise location and orientation as shown in figure 11a. The assistant then initiates the teaching sequence and moves the arm to pick-up-location of each tool. The moving starts by releasing the robotic arm to freedrive mode[2]

[2]In freedrive mode, a human user can grab the robotic arm and move it to a desired position manually. The robot stays still but does not resist movement if pushed to any direction, unless the arm is approaching a restricted area.

and then grabbing the robotic arm by hand and gently moving it to the desired position and orientation as shown in figure 11b. The location is saved to a database in the robot controller software. A matching virtual 3D model of the tool object is separately generated and stored in the VR software. An example model is shown in figure 11c. When the tool is carefully placed on the table to match the stored 3D model dimensions and orientation, the robot does not need to know exactly what the tool is but it is able to pick the tool up for use as shown in figure 11d. The human operator is left responsible for the correct usage and preventing undesired collisions with the patient and other equipment.

The built-in force feedback function[3] of the robotic arm estimates the force required to perform the movements without external interference and is then able to estimate the force applied to the environment. Hence the robot needs to know the mass of the objects it is carrying. Otherwise the carried mass would cause the force feedback function to provide false information about a downward force that is in reality the weight of the tool. Naturally, the force function itself could be used to estimate the mass of picked up objects. However, since it is based on estimates the result would be less than optimal. Furthermore, the force feedback function of the control system is based on the estimated force function. Thus, estimating the weight of the object would result in decreased accuracy of the force feedback function. Therefore, all the tools must be carefully weighed and the masses stored in the database. Consequently, the UR3 controller is able to estimate the force applied more accurately.

As discussed in section 2, industrial robots working in the same space with humans have to comply to the safety standards. For the tool location teaching part, the robot uses the built in safety functions and the behavior of the robot is predictable. Thus, this part of the system should comply to the safety requirements.

4.3 Recording and replaying tasks in VR

The control of the robotic arm happens by capturing location and rotation data in the VR scene and using the data to control the arm. The location and rotation data combined with timing data can be easily saved for later use as well. Possible uses for the collected and saved data include playback or analysis of the movements.

Replaying the movements could be useful for tasks that are performed identically every time. A user tells the system to start recording, performs a set of movements and tool changes and finally tells the system when to stop recording. Now, the system is able to perform the set of actions repeatedly without a human operator, like an industrial robotic arm performs in mass production factories. In this use case, the operations include live patients that have unique qualities. Consequently, the requirements vary slightly for each operation even if the intent of the operation remains the same. This kind of operations cannot necessarily be performed with exactly similar movements. However, some parts of the operation might be performed without contact to the patient. These parts could be performed with this kind of recorded set of actions.

[3]The force feedback function is explained in detail in section 4.4.

As the exact replaying of a set of movements seems to be disadvantageous, it would be beneficial to extract the purpose of each movement and try to replicate the movement in each situation based on that purpose. Some sort of advanced analysis of the control data would be required to accomplish the extraction. One approach would be to perform and record an operation several times and teach an AI system with the control data. A trained AI might then be able to perform operations autonomously even in varying situations and environments.

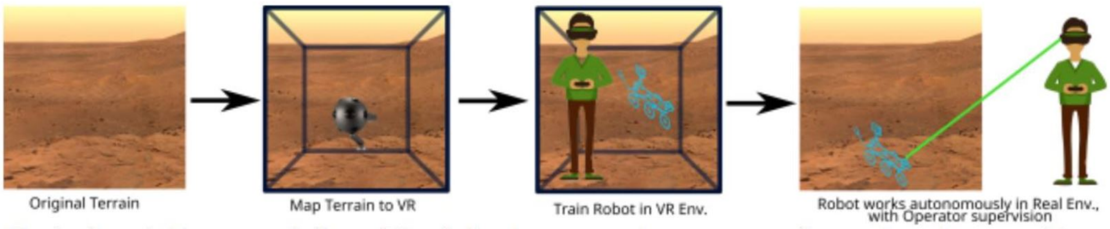


Figure 12: Training an AI to perform autonomously.

Moreover, this particular functionality could be used in various situations outside of the medical setting of this thesis. For example, a robot operating in difficult terrain could be trained with the functionality. The training process is shown in figure 12. First, a robot requests human assistance during operation because of an unfamiliar situation. The terrain around the robot is mapped into a computer generated VR world. Next, a human operator maneuvers the robot through the terrain and the control sequences are recorded. Finally, the robot is able to perform autonomously in that situation with the recorded sequence. Eventually, after repeating the above described process multiple times, the AI propelling the robot might be able to operate completely autonomously in any terrain by applying the previously recorded control sequences in varying terrains. This functionality combined with the AI training possibilities resulted in an invention disclosure within Nokia.

4.4 Force feedback function

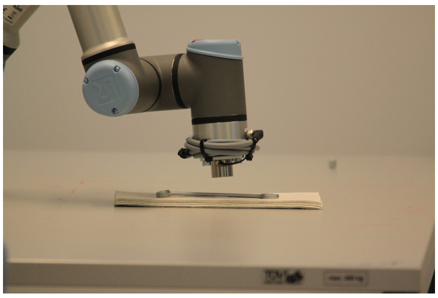
A fundamental part of operating a device remotely is to have sufficient feedback. Without feedback the user might not be able to control the system accurately enough. In the worst case, a lack of feedback could even render the system completely unusable. For example, a user might want to move the arm a specific distance upwards but without feedback the resulting movement might be only half of the desired distance, which could result in collisions.

Providing feedback is not in the focus of this research work. However, the system is able to demonstrate possible ways of providing feedback. Thus, a force feedback loop is developed purely for demonstrative purposes.

As discussed in the hardware section 3.1, the robotic arm is capable of estimating the force magnitude it is applying to its environment. Furthermore, it is able to estimate the direction of the applied force. This force estimation provides sufficient data for providing feedback for the user. However, the problem is not the generated data, but how to deliver it to the user in an intuitive manner. The HTC Vive handheld controllers have a built in haptic feedback device. More accurately, the haptic feedback is a vibration generator. The strength of the vibration can be precisely controlled from the Unity scene via SteamVR API. Controlling the strength of the vibration with the force estimation data from the robotic arm is a simple feedback channel. This functionality is examined in the next two sections.

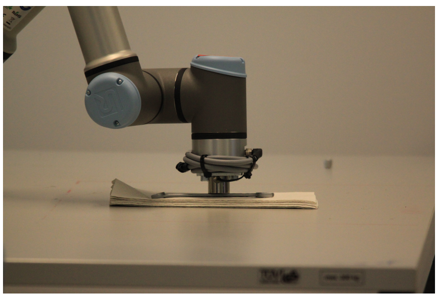
4.4.1 Function operation

Combining the force magnitude data from the robotic arm to the haptic feedback functionality of the HTC Vive, a functional force feedback function is achieved. The functionality level is evaluated with a testing process. The testing process is illustrated in figure 13. First, as shown in figure 13a, a test object is placed on the operating table. This test object is then pushed against the table with the robotic arm using the remote control system. Shown in figure 13b, a user moves the virtual instrument in the VR towards the table. The robotic arm follows this movement until it reaches the object placed on the operating table, as seen in figure 13c. The user continues to move the instrument towards the table, resulting in instructions that the robotic arm should also continue moving towards the table. As the control loop detects that the robot is applying a force, the VR controller starts to vibrate. Subsequently, the user learns that the robot is pushing against something and is able to act accordingly.



(a) A test object on the operation table.

(b) Moving the arm downwards.



(c) The robot pushes against the test object.

(d) VR controller vibrates.

Figure 13: Force feedback testing process.

As discussed in the control loop description, the loop computes a difference between the desired pose[4] and the actual current robot pose. The difference is used to calculate a desired speed for each robot joint servo motor, which is the control signal shown in figure 14. Since the robotic arm is pushing against an object, the difference between desired and actual pose increases. The increased difference results in an increased desired speed signal which is a result of the PI controller. The robotic arm controller has a model which reads the angular position and velocity of each joint and the expected effect of the signal supplied to the joint. By comparing the actual effect of the control signal to the expected effect, the controller is able to estimate the force magnitude and direction that the robotic arm is applying to its environment. For clarification, the force estimation process is shown in figure 14.

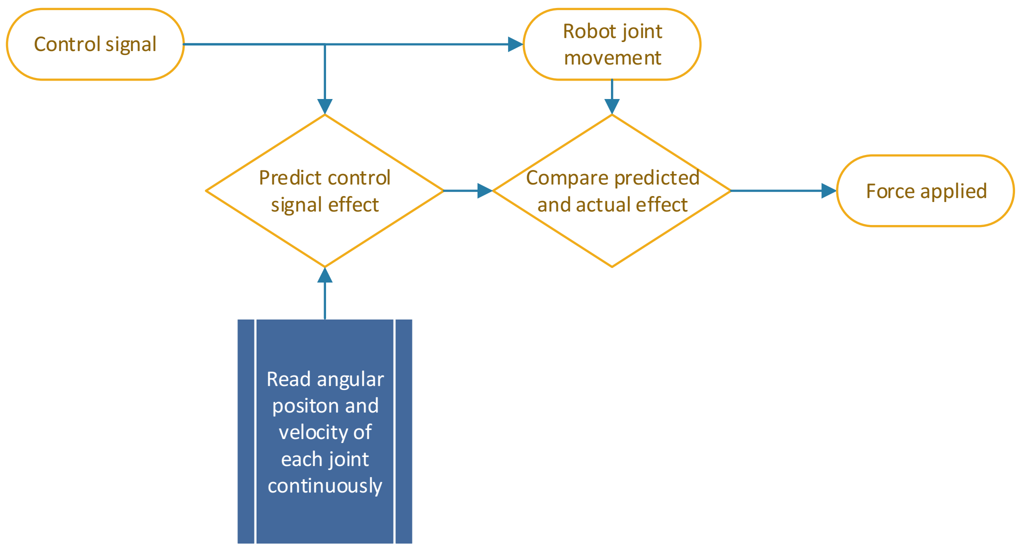


Figure 14: Force estimation process.

The estimated force data is sent back to the VR scene via the network connection. The data is converted to a percentage value with 0 representing no force applied and 100 representing the maximum force. This percentage value is supplied to the haptic feedback vibrator of the controller. The result is a system where the controller in the user's hand vibrates with a strength that is dependent on the force applied in real world. Feedback sensitivity can be adjusted by altering the conversion in the program.

4.4.2 Functionality

The feedback function described above appears to work relatively well. However, the vibration of the hand held controller seems to be an unsatisfactory medium for

[4] A pose describes an exact position for each of the robot joints to achieve a desired position for the robot tool head.

the user to realize the force applied to the environment. Furthermore, the function provides feedback only with rough contacts. The force estimation of the robotic arm built in safety system has a minimum value of 50 N [39]. The force feedback function provides some signal below 50 N but the threshold is not measured.

With this level of feedback, a user might be able to perform tasks that do not require high precision of applied force, but an improved feedback functionality would be required for the system to be actually useful in precise operations. A strain gauge system or some other more precise data source might improve the system in a way that the vibrating haptic device would be sufficient. However, there are other mediums for delivering the feedback for the user. For example, in a VR game called Surgeon Simulator VR: Meet The Medic [2] the user receives visual cues about the force applied. More specifically, should the user try to push against a virtual wall, the bones inside the displayed virtual hand appear to detach and continue inside the wall, while the rest of the hand pushes against the wall as illustrated in figure 15. This kind of visual cue might provide better understanding of the actual applied force. Furthermore, if the user is shown a video feed from the robot site, the developed force feedback function seems to be easier to understand.

Figure 15: Pushing a wall in Surgeon Simulator VR: Meet The Medic [2].

4.5 The results of the first use case

The first use case tested the system in a situation representing an actual medical operation. Even though the use case had some limitations, it soundly tested the functionality.

Overall, the system seemed to work relatively well in moving objects and tools according to the input from the user in VR. A developer of the system was able to control the arm very precisely compared to other less experienced users. For example, a developer was able to pet the stuffed bunny patient gently, while first time users generally managed to accomplish ”more violent contact.” A video of petting the bunny can be seen in [15] and a screen capture of the video in figure 16. The difference might be caused by a combination of aspects that the user must learn in order to achieve more precise control. These aspects may include the behavior of the robot and the control loop, network delay, downscaling of the movements, ambiguous feedback feeling and general familiarity with VR gear.

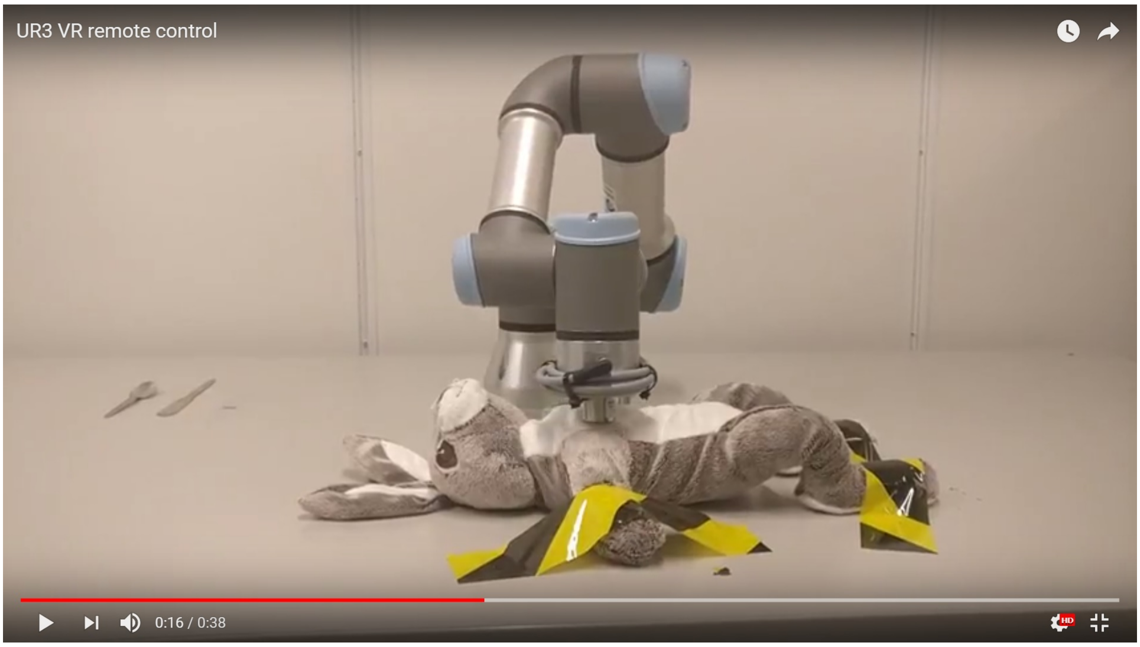


Figure 16: Video: Gently petting the bunny [15] https://youtu.be/bQ99AZRAp5o.

Picking up tools and other objects with an electric magnet appeared to work in this kind of research setting. However, it was unreliable as the orientation of the carried object relied on friction between the magnet and the object. Touching the patient with a tool caused the tool occasionally to rotate around the electric magnet which then caused discrepancy between the virtual model of the tool and the actual real life tool. Furthermore, the electric magnet was a bit limited since it was unable to pick up non steel objects without a steel attachment on them.

Recording and replaying the exact movements retained the accuracy of the movements with all the limitations of the control loop. Since the control loop excluded a timing check, the movements could be replayed in altered speed. Therefore, a timing check would be necessary in applications where the speed of each movement is crucial. Teaching an AI with recorded movement was not tested in practice, which is a subject for further research.

The force feedback function presented the biggest deficiency in this use case. While the force feedback proved that such functionality is possible to implement, it failed in providing usable feedback. Improved feedback could include strain gauges or other sensors as well as video feedback and other visual cues. Further research is clearly required.

While the use case was examined from a medical viewpoint, the functionality seemed to be very similar to many other possible applications. There are numerous tasks that are conducted by picking up a tool and moving it around to accomplish something. This system could be effortlessly modified to suit other applications as well. Especially applications where the environment or the tasks would be dangerous for a human worker would be most beneficial use cases for this kind of system.

5 Second use case: Remote observation and diagnosis

In this section, the second use case is described in detail. First, the motivation for this use case is discussed and then the use case is examined from the perspective of the used subfunctionalities. The subfunctionalities presented in this use case overlap in some parts with the first use case. In the end of the section, the remarks and the security implications made in this use case are reviewed.

5.1 Motivation

VR is a concept in which the user is immersed completely in the virtual world and minimum amount of real life elements are presented. In the first use case, the user works with the virtual model of a patient and the only real life element is the force feedback function. Definitely the user is aware that the operations performed in the VR are replicated in the real world, but the information flow is one-way. On the contrary, augmented reality is an adjacent concept in which the real world is augmented with virtual elements. Moreover, mixed reality combines these both concept in a more fluid way.

Working remotely through a medical operation requires the real life elements of the patient to be transferred back and displayed to the doctor. Since this thesis is researching medical operations performed with VR, a natural part is to research the possibilities of displaying these necessary elements. It is plausible that a doctor might be unable to perform any operations without seeing the real world patient in some manner. Performing medical operations in real life has the natural element of looking around and observing many things. This could be called visual feedback from the situation, which helps in conducting the operation. Furthermore, a substantial part of the equipment in medical establishments consists of different sensors that help the medical staff to monitor a patient during an operation.

In the second use case, the visual feedback is taken into focus. It might be possible to create a virtual model of the patient and update it in real time to provide visual feedback. However, a video feed from the remote patient site provides visual feedback without advanced equipment required in the generation of a virtual updating model. Consequently, the robotic arm is equipped with a video camera for this purpose. The main drawback is that the robot is now equipped with a camera and it is not able to pick up and use tools at the same time. As a result, the use case is restricted to making observations, which might enable a doctor to make diagnoses remotely. Remote diagnosis might be especially beneficial in emergency situations where immediate actions might save the patient.

Displaying the recorded visual feedback in the VR is ambiguous. In the real life, a doctor uses their own eyes and head movements to achieve different viewing angles. Furthermore, a doctor could use an endoscope or different imaging technologies to have even more options in viewing angles and penetration of the view. In this use case, two replications of these real life possibilities are examined. First, the robotic arm is made to follow a virtual instrument in the VR, which is similar to the first use case. The video feed is displayed on a virtual screen next to the virtual instrument, which feels like using a video camera except that the video displayed is coming from the real world site. This vaguely represents the usage of an imaging device in real life and is illustrated in figures 17a and 17c. Second, the robotic arm is set to follow the user's head. In this case, the video feed is displayed directly in front of the user's eyes, which creates a feeling of actually looking around in the real world site. This represents the doctors ability to normally observe a patient and is illustrated in figures 17b and 17d.

In addition to the video feedback function, the two site setup of the system is examined in more detail. The system utilizes a network connection in transferring the data between two remote sites. Future networking technologies enable this kind of connectivity reliably over internet network, which removes the need of a dedicated connection between the two sites. Some aspects of the network functionality, safety and security are discussed.

Apart from the network safety and security, the system can also demonstrate more physical security functions. Many advanced systems have several means for preventing unauthorized usage. Possible approaches include:

- physically securing the system with restricted access to the system premises
- controlling the usage with password or other authentication in the system interface
- surveillance
- a combination of different approaches

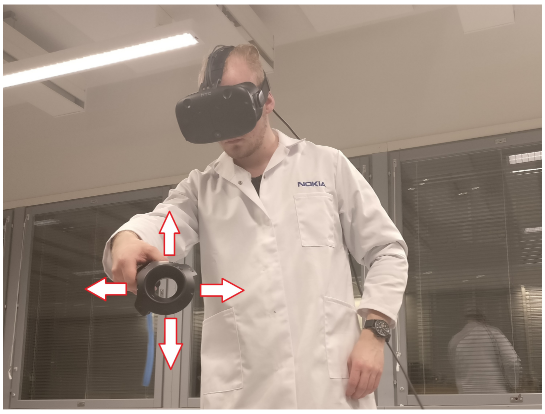
This system is also protected with passwords and locked doors. However, these approaches limit the authorization to the starting point of the system operation. For example, an authorized user might be interrupted while they are using the system. Wearing a HMD isolates the user from the events happening in the real world and leaves the user vulnerable for intrusion. The VR gear and environment provide some unique data points that can be utilized in enabling more advanced authorization. These possibilities are discussed in section 5.4.

5.2 Video application

While working remotely the user may need to look around for many purposes. To fulfill this need a video camera is attached to the end of the robotic arm with a magnet. The video feed is transferred to the VR computer via a USB cable. Naturally, this would not work in remote operations so an internet transfer channel would be required. Two different implementations are researched next.

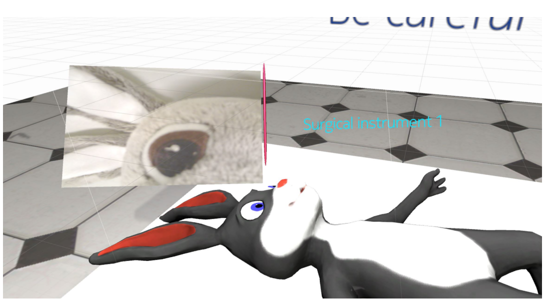
In the first implementation, the functionality of the first use case is modified for this purpose. A virtual screen is added next to the instrument to display the video feed as shown in figure 17c. The user is able to move the instrument freely in the VR with their hand as shown in figure 17a. The position and rotation of the instrument is sent to the robotic arm which then replicates the movement. Since the virtual screen is next to the instrument, the interface feels like an observation tool. The user could use this kind of interface for making precise observations. As an example, a doctor could be able to look into the patient's eyes.

In the second implementation, the scene is modified even further. In this version, the system tracks the movement and rotation of the user's head as shown in figure 17b. A virtual screen displaying the video feed is placed directly in front of the user's face as seen in figure 17d. As a result, the user feels like they are standing on the operating table. Furthermore, the user is able to look around quite freely and even move around in a limited space. This kind of interface might be more suitable for getting a more comprehensive image of the situation on the operation table. Moreover, the ability to move and look around might be useful for monitoring the work of other people. For example, a doctor could remotely oversee a student performing an operation in a teaching situation.



(a) Hand control.

(b) Head control.



(c) Hand control view in VR.

(d) Head control view in VR.

Figure 17: Different control mechanisms and respective views in VR.

5.3 Two-site setup and remote operation

The fundamental idea behind this prototype system is to enable remote operation of a robotic arm. Consequently, a communication link between the control site and the operation site is needed. In this section, the connectivity is examined in detail. The actual data delivered is described in more detail in the control loop section 3.4.

The VR site has a personal computer that is used for running the VR scene and the HTC Vive hardware. The operation site has the controller device of the robot, the robotic arm and a personal computer as well. The PCs used are equipped with Ethernet ports as well as wireless Wi-Fi connectivity. The robotic arm controller device is equipped with an Ethernet port for communications.

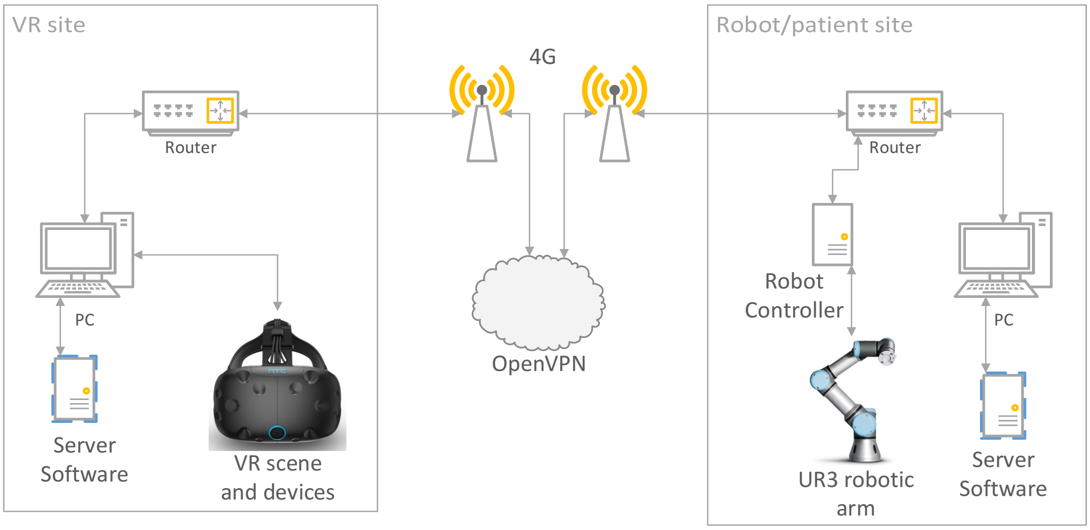


Figure 18: Network setup diagram of the prototype system.

In the first phase of the development, the robotic arm controller is connected directly to the VR computer with an Ethernet cable. Transmission Control Protocol (TCP) [3] is used for the communication, since it is crucial that the data is delivered in correct order and no data is lost in the delivery. In more detail, the VR scene actually sends data to a server software that forwards the data to the robotic arm. The robot controller receives the data and responds with other data, which the server software then forwards to the VR scene. Each of these links uses TCP. This setup appeared to work especially smoothly and any delays in the system are virtually indistinguishable.

While an Ethernet cable provides somewhat remote operation, a wireless network is required for the actual remote operation. Consequently, a network setup shown in figure 18 is installed. The robotic arm controller device is connected to a separate PC running server software. Now, the server software running on the VR computer forwards the data from VR scenario to that PC over a 4G connection. The VR site and robot/patient site handle all traffic using TCP to provide reliable data transfer within the sites. The communication between the two sites uses User Datagram Protocol (UDP) [3] over an OpenVPN running in cloud. A demonstration of the prototype system remote operation was given in DIMECC Cybertrust program final seminar. A video of the demonstration is available at [42] and a screen capture of the video in figure 19. In the video, the distance between the robot/patient site and the VR site is about 10 km and the remote operation remains smooth and network delay reasonable.

Figure 19: Demonstrating the prototype system remote operation [42] https://youtu.be/k9bYKdxGVDA.

5.3.1 UDP for network problem and attack simulation

Using UDP between the two sites enables simulation of different network problems and attacks because the protocol only broadcasts data and does not check whether the data is delivered correctly.

To test this, a distributed denial of service (DDoS) attack is simulated to test how the system behaves under such attack. A DDoS is an attack where the communication link is overloaded with malicious traffic with the intent of blocking any useful communication. A successful DDoS attack causes loss of packets and delay in delivery. [35] Because the system uses UDP in the communication, the lost packets are really lost forever. With TCP the lost packets would be sent again until they are delivered.

Since UDP does not include any means for ensuring data delivery or correct order, a simple order check is applied. Each message sent includes an incremental counter. When receiving a message, the server always checks that the counter is greater than the one in previously received message. In case the counter is lesser, such as in the case of a lost packet, the message is ignored. This acts as a filter that blocks the messages that are older than the previously received messages. Consequently, the operation of the robot is not as smooth, since there is always some packet loss and disordered packets with UDP even without any attack. The data flow and processing were examined in more detail in section 3.4 and in figure 7.

The attack appears to cause severe jitter[5] and delays to the control loop. The

[5]Jitter refers to variance in packet delivery delay. In this case it causes shaky movement due to inconsistent order of desired position data.

effects are not measured, but accurate usage of the system under such attack is impossible. These are the desired effects that Nokia wants to research with this system and potentially create solutions to securely enable this kind of remote operations over Internet connection. For example, Nokia has already developed a deep learning based system that is able to differentiate attacks from regular network traffic and consequently filter out the attack [13]. This attack mitigation appears to successfully retain the system usable under a simulated DDoS attack and keep network delays reasonable.

Online games constantly face the same problem of network delays and packet losses. Fortunately, game industry researchers have found an effective mitigation to the problem. The behavior of a player is constantly analyzed and an extrapolation is calculated. Should the network undergo some problem, and the player's commands lost for a moment, the game engine assumes that the player behaves according to the extrapolation and thus the game is able to continue without interruption. For example, if a player was running forward in the last received data packet, it is assumed to continue running forward until the next data packet is received i.e. the velocity of the player is assumed to stay constant. The method is called dead reckoning [31]. The HTC Vive actually uses dead reckoning to improve the tracking data of the HMD and the controllers [14]. The same method could be incorporated in the robotic arm control loop as well to improve the behavior under attack and network problems.

5.4 User identification and other security aspects

The system is developed with the intent of researching security aspects related to future remote operations in medical field. Once the desired remote operation functionality is achieved, the focus is shifted towards security.

Such advanced systems generally require some kind of authorization of the user. Traditional approaches include credentials, passwords, fingerprint sensors and physically secured working spaces among other solutions. However, the VR environment provides unique possibilities to accomplish the required security level. One of these possibilities was studied in this research work.

As discussed in the hardware section, the HTC Vive provides relatively accurate tracking of the user's head and hands. Analyzing this tracking data, three types of signals can be found.

First, the user is performing a movement. By instructing the user to perform specific movements, this signal could be isolated. Furthermore, a database of typical movements can be provided with which the system could be able to isolate a set of movements out of the received tracking data.

Second, the HTC Vive system uses sensors that produce some sort of sensor noise signal. The sensor noise can be usually modeled and filtered out.

Third, the user causes some noise to the signal, such as unconscious shaking of the hand. This noise may have different sources such as unique features in the user's body, level of experience, alertness or exhaustion. If first and second signals are filtered out of the tracking data, the resulting signal would include only the third signal. Hypothetically, the third signal might be user specific. This kind of user specific signal could be used for identifying the user. Furthermore, this could result in a continuous identification if the system is capable of filtering out the first and second signals in all situations.

Having a continuous identification functionality would secure the system in situations where traditional approaches would fail. For example, a doctor might be interrupted by someone and forced to perform malicious movements. Hypothetically, the user specific noise pattern varies over the course of an operation. The system might be able to monitor the noise pattern and raise an alert if the noise pattern is atypical. In addition to an intruder, this could be a result of exhaustion, influence of some substance or even lack of experience.

5.5 The results of the second use case

In the second use case, the VR control system was used for softer medical operations such as observations and measurements. A video feedback system was developed and two different control mechanisms were tested: control by hand or head movement. The robotic arm was equipped with a video camera and the video feed was displayed in the VR scene. This video feedback system appeared to provide useful means for making observations in medical operations.

By controlling the robotic arm with hand, very precise viewing positions and angles were achieved. The control felt like using an imaging device in real life. On the contrary, when controlling the arm with head movement the control was not so precise. However, this control mechanism appeared to provide an intuitive means for getting a more holistic view of the situation on the operation table. Indeed, when the arm followed the movement of the user's head and the video feed was displayed directly in front of the face, the resulting experience was similar to standing on the operation table and looking around.

The first use case relied only on force feedback from the robotic arm and the user saw only a virtual model of the patient. However, the force feedback function appeared to be insufficient for performing many operations. Thus, the first use case could be improved if the user had visual feedback from the patient side combined with the force feedback. This could be achieved with the system used in the second use case.

The video feedback could be improved further with a higher quality camera device. Moreover, the HMD is capable of displaying stereo video and thus especially the head movement control mechanism could be greatly improved with the addition of a stereo video camera. These are subjects for further research, since these aspects were not covered due to resource limitations.

The remote connectivity setup was also examined in detail. The communication over a 4G connection seemed to be sufficient for basic usability. Network delay was not measured, but based on use case experiences the delay was insignificant when the system was operating under optimum network conditions. In previous research work, where Internet was used for communication in similar robotic medical system, lag times ranging from 30 to 150 ms were completely undetectable by surgeons [22], which is in line with the experiences with this prototype system.

The reason why the prototype system was developed in the first place was to research security issues. The simulated DDoS attack completely revealed the vulnerability of remote robotic surgeries and that there is need for research in that area. Furthermore, the prototype system proved to be capable for researching those aspects. Also Dr. Smith mentions that the internet technology is already sufficient for remote surgeries, but there are still major obstacles such as internet resilience, security and liability issues [22].

VR appeared to present some unexpected means for providing additional security in the system. Some concepts were already explored with the system, such as continuously identifying the user. Additionally, the system might be able to detect if the doctor is exhausted or under influence of some substance. However, further research is required.

6 Summary and Conclusion

In this thesis, VR technology was utilized in developing an intuitive control interface for an industrial robotic arm. The motivation for using this kind of control system is to enable remote operations in medical field. Possible medical operations include observing symptoms, diagnosing and performing operations such as surgeries. Performing medical operations remotely might benefit the patient in multiple ways. For example, a patient might not need to travel to meet a specialized doctor to get a diagnosis and treatment, which might hasten the beginning of the treatment. This can potentially enable more inexpensive medical treatment as well as save lives in emergency situations.

Medical operations are mission critical scenarios where the system is required to operate reliably and safely under every situation. This aspect was one of the reasons why medical operations were chosen as the subject for the use cases in this research. The research was conducted as a part of cyber security research in Nokia and the resulting system will be used for further network security and safety research. Some security aspects were researched also in this thesis.

The utilized VR equipment was HTC Vive, which comprises of a HMD, two handheld controllers and two lighthouses for tracking. The HTC Vive system includes software that takes care of displaying the VR view and tracking the user movements. With the equipment and software the user gets immersed into the VR and is able to move around and interact with virtual objects in the VR scene. The controlled robotic arm was a Universal Robots UR3 industrial robot. UR3 has six DoF and thereby is able to replicate movements of a human hand. It is also a so called collaborative robot, which has the necessary safety functions for operating in a shared space with humans as required in EN ISO 10218-1:2011 safety standard and ISO/TS 15066:2016 technical specification [11, 12].

The system was divided into two sites: VR site and robot/patient site which were connected with a network connection. The doctor site had a medical setting built in a VR scene, representing an operation table with an virtual bunny as a patient. The doctor was able to observe the virtual patient and use virtual instruments to perform operations. The movements of the doctor were captured and sent to the patient site. At the patient site, a robotic arm was mounted on an operation table. The robotic arm replicated the movements of the doctor and, for research purposes in this thesis, had a stuffed bunny as a subject of the operations.

As stated, the two sites were connected with a network connection. More specifically, the computers at each site were connected to a cloud server using 4G connection. The VR computer had a separate server software that received the data from the VR scene and used UDP protocol to forward the data to a server PC at the patient site via the cloud. At the patient site, the server PC received the data, filtered out packets that were in incorrect order and forwarded the data to the robotic arm.

Industrial robots are designed to function in a factory environment in which the operations are programmed in advance and strictly repeated in the process. In case of a change in the environment or an unexpected situation, industrial robots usually require reprogramming or other human interaction to be able to continue operation.

As such, the control software of UR3 does not support a continuous control input without predetermined locations and movement paths. A custom control loop was developed to accomplish the desired control which was able to replicate human hand movements intuitively in real-time.

Once the control system was established, the functionality of the system was studied in two use cases that represent different medical operations. In the first use case, the robot was used to move tools and objects around, representing a situation where a doctor performs some operation to the patient. An electric magnet was used for picking up the tools and a teaching sequence was developed with which an assistant could teach the robot where to pick up the tools. A force feedback loop was also developed to help the user control the arm better. Additionally, the system was used to record and replay movements. The recorded movements could be used for training an artificial intelligence to perform tasks autonomously. This invention led to an invention disclosure within Nokia. The remarks made in the first use case were the accurate replication of the movements with a delay of under 200 ms, insufficiency of the force feedback function and potential uses in machine learning applications.

In the second use case, lighter medical operations, such as measurements and observations, were examined. Observing a patient and diagnosing requires different feedback from the patient site. For this purpose, the robotic arm was equipped with a video camera. The video feed was sent back to the VR scene, in which the doctor is able to control the robotic arm and thus the viewing angle of the camera. Two different control mechanisms were tested, hand control and head control. Controlling the viewing angle with hand appeared to work for precise observations such as looking into the patient's eye. Controlling the viewing angle with head movement provided the doctor a more comprehensive picture of the patient and the operation table, but it seemed to work poorly for more precise control.

The security aspects of the system were examined in the second use case as well. The communication link between the two sites was attacked with a simulated denial of service attack. The system rendered completely unusable under such attack. Nokia security research team has developed a machine learning based mitigation that is able to detect normal traffic patterns and prioritize the normal traffic over the attack traffic. This mitigation was able to retain the system usability with only slightly increased network delay under attack. Additionally, the control system is able to provide even more advanced security features. By isolating user specific unconscious variances in movements, the system might be able to identify the user continuously. Furthermore, the system might be able to detect if the user is interrupted during the operation and prevent accidents by blocking the robot control in these situations.

As a conclusion, the system serves as a proof-of-concept that this kind of VR control system can be used for simulating remote medical operations, which was one of the goals of this thesis. Especially lighter medical operations such as observations and measurements are close to reality with this system. For example, x-ray or ultrasound imaging could possibly be conducted with the system at this functionality level. However, there are several aspects that require further development before the system is adequately advanced for real medical use. The VR equipment was the limiting factor in the accuracy of the system, since the tracking brought noise and some inaccuracies into the system. The custom control loop functioned sufficiently with slow movements, but rapid movements caused the robot to replicate the movements inaccurately. Moreover, the developed force feedback function did not provide sufficient feedback for precise control and would need improvement and augmentation with other feedback mediums, such as visual feedback.

Additional remarks outside of the primary research objectives were also made during the development work. Combining systems and software from different sectors of engineering seemed to bring additional challenges in the development. This is a result of differences in the tools and processes used in each field. For example, the 3D models used in the game building platform Unity are distinctively different from the ones used in mechanical engineering and robotics field. Importing models from the software tools from one field to another requires work and specialized software. This is reasonable, because different applications have their own requirements for the models. However, this phenomena also adds complexity in combining the technologies like in the case of this thesis.

The system was used for relatively brief stretches of time while conducting this research work. However, continued use of the HTC Vive equipment could possibly have negative effects on the performance of the doctor. Some users even experience VRISE while using VR HMDs which might be an issue for general use of this system. The VRISE was not specifically studied in this thesis, but it appeared to be a minor issue. VRISE was practically nonexistent especially for users with previous experience using VR HMDs.

The ease of use and adaptation to the system appeared to vary wildly between users. Some users required constant assistance to be able to use the system while some users adapted in a few moments. This might be a problem of the VR gear, but also of the VR scene that was built for the system. Further research would be required to understand why this occurs.

Improving the VR environment further might enable other possibilities that were completely ignored in this thesis. It might be possible to exclude some unwanted features of the real life environment and retain or improve the features that are necessary for conducting the tasks at hand. For example, the movements of the doctor can be scaled down, which could enable much more precise operations. Even further, the time frame could potentially be altered in a way, that the doctor could perform movements slower in VR to enhance accuracy, but the actual movements would be performed faster. Some research in the field is conducted with the purpose of providing medical care in dangerous environments. For example, the lag time research conducted at Florida Hospital Nicholson Center is said to have funding from U.S. Department of Defense [22]. Thus, there seems to be demand for medical teleoperations in military applications where the dangers of war zones obviously could be mitigated for the doctors at the VR site.

Another direction for further research could be educational uses of the prototype system. Previous research with VR techonology has shown that surgical training performed in VR environment can improve operation room performance significantly for surgical residents [29]. Based on the experiences in the two use cases, the prototype system could potentially be used for training of surgeons or physicians.

Apart from the intended network safety and security aspects, there are multiple feasible improvements for the system. There is already some research of so called haptic gloves, that can provide even more intuitive interface for tracking a human hand and its gestures and finger movements. These gloves might also provide means for improving the inadequate force feedback function of the system. Other improvements could include better end effectors and further optimization of the arm behaviour in rapid movements.

VR technology is currently advancing at an accelerated phase. There are new standalone systems coming to market that enable similar VR experience without an external computer and the cabling to the HMD. The accuracy and display systems are improving likewise. Applying these newer VR technologies is an obvious improvement for the system.

The automation aspects were also left out of the scope of this thesis. However, an evident subject for further research and improvement of the system would be developing automation in the medical operations. For example, the prototype system could be used for automatically observing a patient.

Despite its deficiencies, the system can be used to research the security and safety aspects in mission critical remote operations, which was the major goal of this thesis. Some security aspects were already examined in the thesis and related invention disclosures were made. The prototype system might not end up to the development of a commercially distributed medical device. However, it provides a platform for research in areas which will be crucial in future network applications.

References

[1] Bethesda Game Studios. *Fallout 4 VR*. Dec. 2017. URL: http://store.steampowered.com/app/611660/Fallout_4_VR/.

[2] Bossa Studios Ltd. *Surgeon Simulator VR: Meet The Medic*. Apr. 2016. URL: http://store.steampowered.com/app/457420/Surgeon_Simulator_VR_Meet_The_Medic/.

[3] R. Braden. *RFC 1122 Requirements for Internet Hosts - Communication Layers*. Internet Engineering Task Force. Oct. 1989. URL: http://tools.ietf.org/html/rfc1122.

[4] DIMECC. *DIMECC Cyber Trust Program*. [online; retrieved 22-October-2017]. 2017. URL: https://cybertrust.fi/.

[5] Enscape GmbH. *Enscape3D*. [online; retrieved 07-November-2017]. 2017. URL: https://enscape3d.com/.

[6] Google Inc. *Tilt Brush*. [online; retrieved 07-November-2017]. 2017. URL: https://www.tiltbrush.com/.

[7] Rayna Hollander. *Here's why AR and VR are set to take off*. [online; retrieved 05-January-2018]. Aug. 2017. URL: http://www.businessinsider.com/ar-vr-2017-8?r=US&IR=T.

[8] Olivia Holmes, Martin S. Banks, and Hany Farid. "Assessing and Improving the Identification of Computer-Generated Portraits". In: *ACM Trans. Appl. Percept.* 13.2 (Mar. 2016), 7:1–7:12. ISSN: 1544-3558. DOI: 10.1145/2871714. URL: http://doi.acm.org/10.1145/2871714.

[9] HTC Corporation. *Vive^{TM} Room-Scale VR*. [online; retrieved 6-January-2018]. URL: https://www.vive.com/eu/product/.

[10] Alejandro G. Iñárritu. *Carne y Arena*. 2017.

[11] ISO, Geneva, Switzerland. *ISO 10218: Robots and robotic devices – Safety requirements for industrial robots – Part 1: Robots*. Standard. 2011.

[12] ISO, Geneva, Switzerland. *ISO/TS 15066: Robots and robotic devices — Collaborative robots*. Technical Specification. 2016.

[13] Aapo Kalliola et al. "Learning Flow Characteristics Distributions with ELM for Distributed Denial of Service Detection and Mitigation". In: (May 2018), pp. 129–143.

[14] Oliver Kreylos. *Lighthouse tracking examined*. [online; retrieved 13-September-2017]. May 2016. URL: http://doc-ok.org/?p=1478.

[15] Heikki Laaki. *UR3 VR remote control: Petting a bunny*. [online; retrieved 08-January-2018]. Jan. 2018. URL: https://youtu.be/bQ99AZRAp5o.

[16] Heikki Laaki. *UR3 VR remote control: Tracking demo*. [online; retrieved 08-January-2018]. Jan. 2018. URL: https://youtu.be/Sfv-Mot8cvo.

[17] Heikki Laaki et al. "Augmenting virtual worlds with real-life data from mobile devices". In: (Mar. 2010), pp. 281–282. ISSN: 1087-8270. DOI: 10.1109/VR.2010.5444764.

[18] Jacques Marescaux et al. "Transatlantic robot-assisted telesurgery". In: *Nature* 413.6854 (2001), pp. 379–380.

[19] Kim Mathiassen et al. "An Ultrasound Robotic System Using the Commercial Robot UR5". In: *Frontiers in Robotics and AI* 3 (2016), p. 1. ISSN: 2296-9144. DOI: 10.3389/frobt.2016.00001. URL: https://www.frontiersin.org/article/10.3389/frobt.2016.00001.

[20] Emma McDonald. *The Global Games Market 2017 | Per Region Segment*. [online; retrieved 05-January-2018]. Apr. 2017. URL: https://newzoo.com/insights/articles/the-global-games-market-will-reach-108-9-billion-in-2017-with-mobile-taking-42/.

[21] T. Liam McDonald. "Are we having virtual fun yet?" In: *PC Gamer US* 1.3 (Aug. 1994), pp. 44–49.

[22] Lucas Mearian. *Hospital tests lag time for robotic surgery 1,200 miles away from doctor*. [online; retrieved 05-January-2018]. May 2015. URL: https://www.computerworld.com/article/2927471/healthcare-it/robot-performs-test-surgery-1200-miles-away-from-doctor.html.

[23] Bishoy Morris. "Robotic surgery: applications, limitations, and impact on surgical education". In: *Medscape General Medicine* 7.3 (2005), p. 72.

[24] Saeed Niku. *Introduction to robotics*. 2nd ed. John Wiley & Sons, 2010. ISBN: 978-0-470-60446-5.

[25] Nokia. *White paper: 5G for Mission Critical Communication*. 2016.

[26] Oculus VR. *Oculus Rift*. [online; retrieved 08-January-2018]. URL: https://www.oculus.com/rift/.

[27] Ole Ravn, Nils Axel Andersen, and Thomas Timm Andersen. *UR10 Performance Analysis*. 2014. URL: http://orbit.dtu.dk/ws/files/105275650/ur10_performance_analysis.pdf.

[28] RoboticOncology.com. *Benefits of Robotic Prostate Cancer Surgery*. [online; retrieved 05-January-2018]. URL: http://www.roboticoncology.com/robotic-prostate-surgery/.

[29] Neal E Seymour et al. "Virtual reality training improves operating room performance: results of a randomized, double-blinded study". In: *Annals of surgery* 236.4 (2002), p. 458.

[30] Sarah Sharples et al. "Virtual reality induced symptoms and effects (VRISE): Comparison of head mounted display (HMD), desktop and projection display systems". In: *Displays* 29.2 (2008). Health and Safety Aspects of Visual Displays, pp. 58–69. ISSN: 0141-9382. DOI: https://doi.org/10.1016/j.displa.2007.09.005. URL: http://www.sciencedirect.com/science/article/pii/S014193820700100X.

[31] Jouni Smed, Timo Kaukoranta, and Harri Hakonen. "Aspects of networking in multiplayer computer games". In: *The Electronic Library* 20.2 (2002), pp. 87–97. DOI: 10.1108/02640470210424392. URL: https://doi.org/10.1108/02640470210424392.

[32] Jonathan Steuer. "Defining Virtual Reality: Dimensions Determining Telepresence". In: *Journal of Communication* 42.4 (1992), pp. 73–93. ISSN: 1460-2466. DOI: 10.1111/j.1460-2466.1992.tb00812.x. URL: http://dx.doi.org/10.1111/j.1460-2466.1992.tb00812.x.

[33] Tesla. *Inside Tesla 06.12.12 - Paint*. [online; retrieved 12-January-2018]. June 2012. URL: https://youtu.be/vXntSUtoEFc.

[34] Alexandra Thompson. *There's a patient under there somewhere: The incredible medical robots that have saved a record number of men from deadly prostate cancer*. Nov. 2017. URL: http://www.dailymail.co.uk/health/article-5114551/Robot-surgery-save-men-prostate-cancer.html.

[35] United States Computer Emergency Readiness Team. *Security Tip (ST04-015)*. [online; retrieved 05-January-2018]. Nov. 2009. URL: https://www.us-cert.gov/ncas/tips/ST04-015.

[36] Unity Technologies. *Unity*. [online; retrieved 6-January-2018]. URL: https://unity3d.com/unity.

[37] Universal Robots A/S. *The URScript Programming Language*. 3.3.4. [online; retrieved 29-September-2017]. Dec. 2016. URL: https://www.universal-robots.com/download/?option=22197#section21968.

[38] Universal Robots A/S. *UR3 robot*. [online; retrieved 05-January-2018]. URL: https://www.universal-robots.com/products/ur3-robot/.

[39] Universal Robots A/S. *User Manual UR3/CB3*. 3.3.3. [online; retrieved 29-September-2017]. 2016. URL: https://www.universal-robots.com/download/?option=21983#section21970.

[40] ValveSoftware. *openvr*. [online; retrieved 07-January-2018]. URL: https://github.com/ValveSoftware/openvr/wiki/API-Documentation.

[41] John Velasco. *HTC Vive Review*. [online; retrieved 6-January-2018]. Sept. 2016. URL: https://vrsource.com/htc-vive-review-3443/.

[42] Venimis Com. *Cybertrust seminar - Yoan Miche - Nokia Bell Labs*. [online; retrieved 17-January-2018]. Sept. 2017. URL: https://youtu.be/k9bYKdxGVDA.

[43] Virtalis Limited. *VR4CAD*. [online; retrieved 07-November-2017]. 2017. URL: https://www.virtalis.com/vr4cad/.

[44] Wachowski Brothers, The. *The Matrix*. 1999.

[45] Weiss Robotics GmbH. *Smart Gripping Solutions for Universal Robots*. [online; retrieved 6-January-2018]. URL: http://www.weiss-robotics.com/gripkit/en/.